PB85-247229

Green River Air Quality Model Development MELSAR - A Mesoscale Air Quality Model for Complex Terrain. Volume 2. Appendices

Battelle Pacific Northwest Labs., Richland, WA

Prepared for

Environmental Protection Agency
Research Triangle Park, NC

Aug 85

U.S. Department of Commerce
National Technical Information Service
NTIS

PB85-247229

EPA/600/8-85/017b
August 1985

GREEN RIVER AIR QUALITY MODEL DEVELOPMENT: MELSAR - A Mesoscale Air Quality Model for Complex Terrain
Volume 2. Appendices

by

K. J. Allwine and C. D. Whiteman
Pacific Northwest Laboratory
Richland, Washington 99352

Interagency Agreement AD89F20970
with the U.S. Department of Energy

Project Officer

Alan H. Huber
Meteorology and Assessment Division
Atmospheric Sciences Research Laboratory
U.S. Environmental Protection Agency
Research Triangle Park, NC 27711

ATMOSPHERIC SCIENCES RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27711

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT DATA (Please read Instructions on the reverse before completing)		
1. REPORT NO. EPA/600/8-85/017b	2.	3. RECIPIENT'S ACCESSION NO. PB85 247229/AS
4. TITLE AND SUBTITLE GREEN RIVER AIR QUALITY MODEL DEVELOPMENT: MELSAR - A Mesoscale Air Quality Model for Complex Terrain Volume 2. Appendices		5. REPORT DATE August 1985 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) K. J. Allwine and C. D. Whiteman		8. PERFORMING ORGANIZATION REPORT NO. PNL-5460 Vol. 2
9. PERFORMING ORGANIZATION NAME AND ADDRESS Battelle, Pacific Northwest Laboratory Richland, Washington 99532		10. PROGRAM ELEMENT NO. CDTA1D/09-0726 (FY-85) 11. CONTRACT/GRANT NO. IAG AD89F20970
12. SPONSORING AGENCY NAME AND ADDRESS Atmospheric Sciences Research Laboratory, RTP, NC Office of Research and Development Environmental Protection Agency Research Triangle Park, North Carolina 27711		13. TYPE OF REPORT AND PERIOD COVERED Final FY-85 14. SPONSORING AGENCY CODE EPA/600/09
15. SUPPLEMENTARY NOTES		

16. ABSTRACT

This volume contains Appendices A through R, which includes 1) instructions for operating the model, 2) program structure and code listings, and 3) summary output listings.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group

18. DISTRIBUTION STATEMENT	19. SECURITY CLASS (This Report)	21. NO. OF PAGES
RELEASE TO PUBLIC	UNCLASSIFIED	366
	20. SECURITY CLASS (This page) UNCLASSIFIED	22. PRICE

EPA Form 2220-1 (Rev. 4-77) PREVIOUS EDITION IS OBSOLETE

DISCLAIMER

The research described in this report has been funded wholly or in part by the United States Environmental Protection Agency through Interagency Agreement AD-89-F-2-097-0 to the Pacific Northwest Laboratory, This document has been reviewed in accordance with U.S. Environmental Protection Agency policy and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

PREFACE

This final report is submitted as part of the Green River Ambient Model Assessment (GRAMA) program conducted at the U.S. Department of Energy's Pacific Northwest Laboratory for the U.S. Environmental Protection Agency. The GRAMA program has, as its ultimate goal, the development of validated air quality models that can be applied to the complex terrain of the Green River Formation of western Colorado, eastern Utah, and southern Wyoming. The Green River Formation is a geologic formation containing large reserves of oil shale, coal, and other natural resources. Development of these resources may lead to a degradation of the air quality of the region. Air quality models are needed immediately for planning and regulatory purposes to assess the magnitude of these regional impacts. This report documents one of the models being developed for this purpose within GRAMA--specifically a model to predict short averaging time (<24 hr) pollutant concentrations resulting from the mesoscale transport of pollutant releases from multiple sources. MELSAR has not undergone any rigorous operational testing, sensitivity analyses, or validation studies. Testing and evaluation of the model are needed to gain a measure of confidence in the model's performance.

This report consists of two volumes. Volume 1 contains the model overview, technical description, and user's guide, and Volume 2 contains the Appendices which include listings of the FORTRAN code.

CONTENTS

Preface..... iii
Figures..... vi
Tables..... vii
Acknowledgments..... viii

Introduction..... 1-1

Appendix A - Maps of major features of the GRAMA modeling region..... A-1
Appendix B - UNIVAC JCL files for compiling and mapping TERRAIN, MET, POLUT, and POLPRC..... B-1
Appendix C - Instructions and FORTRAN code for setting up the base terrain file TOPO..... C-1
Appendix D - Description of input data files available for MELSAR..... D-1
Appendix E - Blank forms for keeping track of files..... E-1
Appendix F - File tracking forms for sample problem..... F-1
Appendix G - Diagnostic messages for MET, POLUT, and POLPRC..... G-1
Appendix H - Summary output listing from TERRAIN for sample problem.... H-1
Appendix I - Summary output listing from MET for sample problem..... I-1
Appendix J - Summary output listing from POLUT for sample problem..... J-1
Appendix K - Summary output listing from POLPRC for sample problem..... K-1
Appendix L - Output concentration tables from POLPRC for sample problem..... L-1
Appendix M - MET program structure..... M-1
Appendix N - POLUT program structure..... N-1
Appendix O - FORTRAN listing of TERRAIN..... O-1
Appendix P - FORTRAN listing of MET..... P-1
Appendix Q - FORTRAN listing of POLUT..... Q-1
Appendix R - FORTRAN listing of POLPRC..... R-1

FIGURES

Number		Page
A-1	Major roads and towns in the GRAMA modeling region.....	A-1
A-2	Prevention of significant deterioration (PSD) Class I areas and the oil shale development areas in the GRAMA modeling region.....	A-2
A-3	Topography in the GRAMA modeling region.....	A-3
A-4	Major rivers and streams in the GRAMA modeling region.....	A-4
A-5	Main river drainages in the GRAMA modeling region.....	A-5
F-1	Summary of TERRAIN runs for sample problem.....	F-1
F-2	Summary of MET runs for sample problem.....	F-2
F-3	Summary of POLUT runs for sample problem.....	F-3
F-4	Summary of POLPRC runs for sample problem.....	F-4
F-5	TERRAIN input/output files for sample problem.....	F-5
F-6	MET input/output files for sample problem.....	F-6
F-7	POLUT input/output files for sample problem.....	F-7
F-8	POLPRC input/output files for sample problem.....	F-8

TABLES

Number		Page
D-1	Description of input data files available for MELSAR.....	D-1
M-1	Subroutine interaction in MET.....	M-1
M-2	Common blocks in MET.....	M-2
M-3	Parameter declarations in MET.....	M-3
M-4	Parameter definitions and values.....	M-4
M-5	Sizing of direct access disk files in subroutines UPPER and SRFC.....	M-5
N-1	Subroutine interaction in POLUT.....	N-1
N-2	Common blocks in POLUT.....	N-2
N-3	Parameter declarations in MET.....	N-3
N-4	Parameter definitions and values.....	N-4

ACKNOWLEDGMENTS

Mr. Alan Huber, the EPA Project Officer, and Mr. Rich Fisher, EPA Region VIII meteorologist, are thanked for their support and encouragement of the work. Mr. Huber and Mr. Fisher have both provided a number of useful suggestions regarding the development of the model and the desired features of the model that will prove useful to the user. These suggestions have resulted in improvements to the model code.

The authors' colleagues at PNL contributed suggestions and criticisms as the work progressed that led to significant improvements in the model's design and in the clarity of the text. Drs. Thomas W. Horst and J. Christopher Doran are especially thanked for their help in that regard. The able programming assistance of Mr. Roger I. Schreck of Pacific Northwest Laboratory is also greatly appreciated.

INTRODUCTION

Pacific Northwest Laboratory has developed a mesoscale air quality model for the U.S. Environmental Protection Agency (EPA) under their Green River Ambient Model Assessment (GRAMA) program. This program was initiated in response to the need for air quality assessment tools applicable in the Green River Oil Shale Formation region of western Colorado, eastern Utah, and southern Wyoming. This region has the potential for large-scale growth because vast energy resouces, especially oil shale, are located in the region.

The model, MELSAR (MEsoscale Location Specific Air Resources model), is a Lagrangian puff model to be applied at long source-to-receptor distances (tens to hundreds of kilometers) and short concentration averaging times (1 to 24 hr). These distances and averaging times are important in estimating the pollutant concentrations at the Prevention of Significant Deterioration (PSD) Class I areas in the region.

Volume 1 contains the overview, technical description and user's guide to the MELSAR model. This volume consists of the appendices to Volume 1, and contains: maps of the region, UNIVAC® Job Control Language files for compiling and mapping MELSAR, descriptions of input data files, computer listings of MELSAR output data, and FORTRAN listings of the program.

®UNIVAC is a registered trademark of the Sperry Rand Corporation.

APPENDIX A

MAPS OF MAJOR FEATURES OF THE GRAMA MODELING REGION

FIGURE A-1

MAJOR ROADS AND TOWNS IN THE GRAMA MODELING REGION

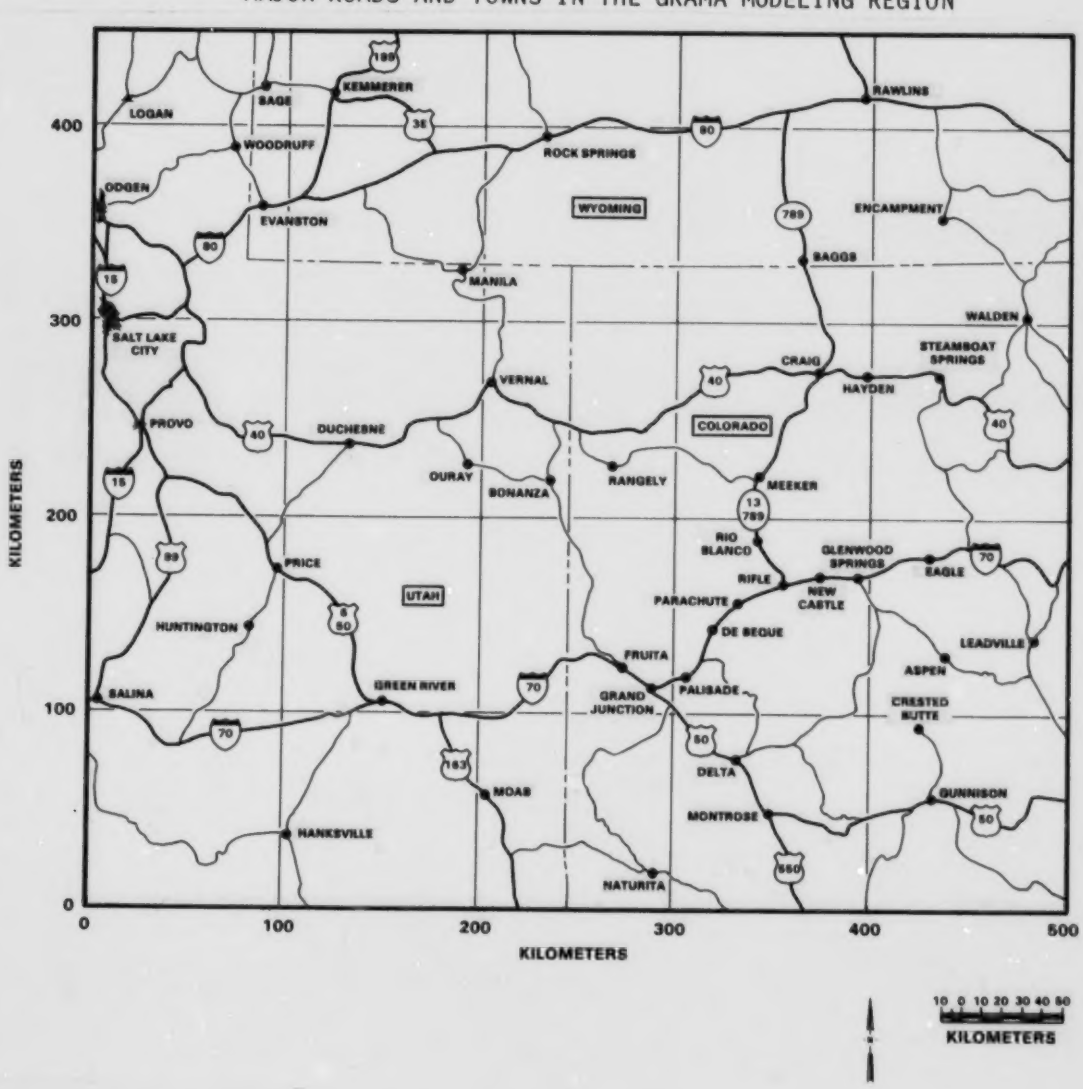


FIGURE A-2

PREVENTION OF SIGNIFICANT DETERIORATION (PSD) CLASS I AREAS
AND THE OIL SHALE DEVELOPMENT AREAS IN THE GRAMA
MODELING REGION

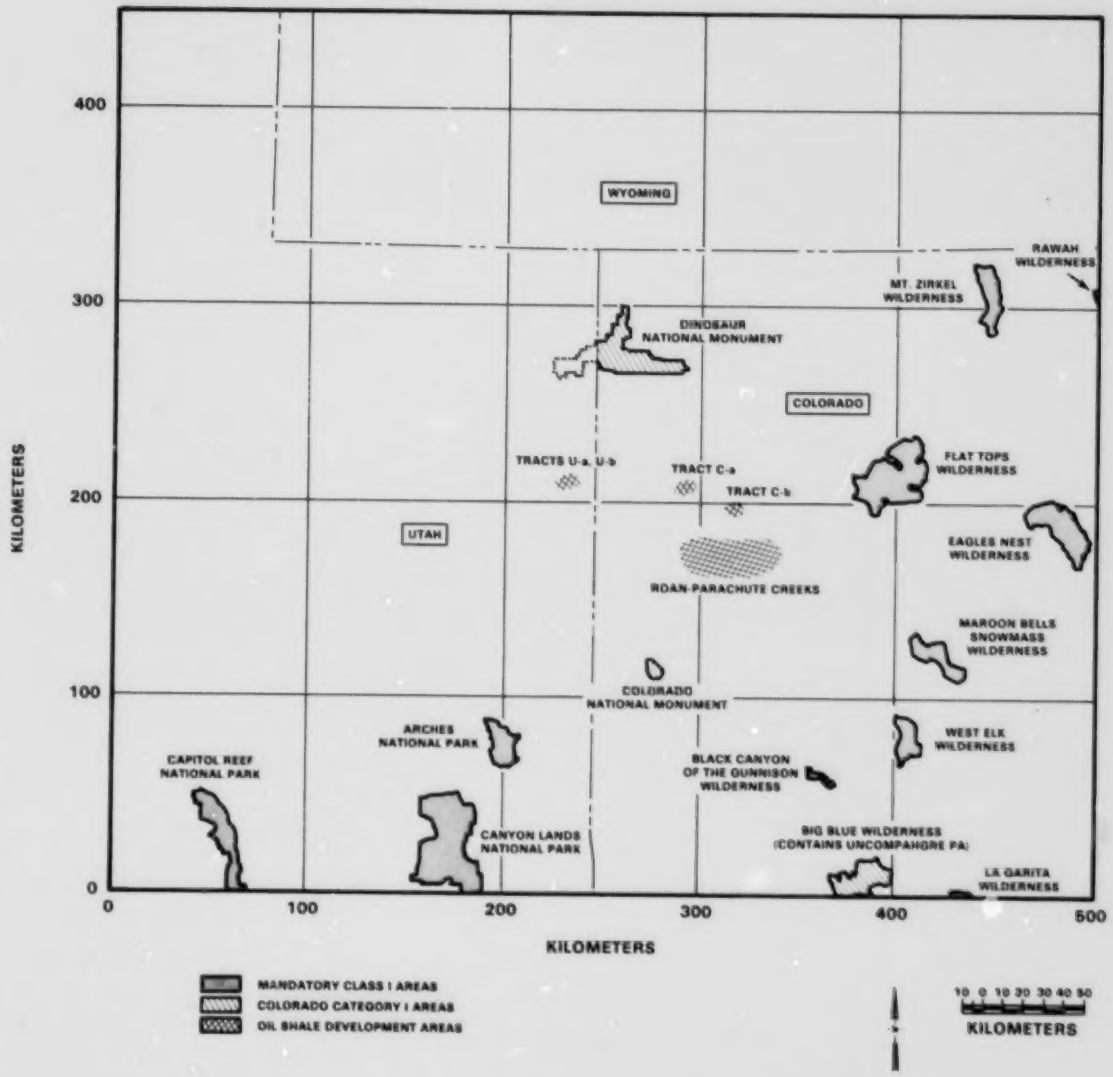


FIGURE A-3

TOPOGRAPHY IN THE GRAMA MODELING REGION

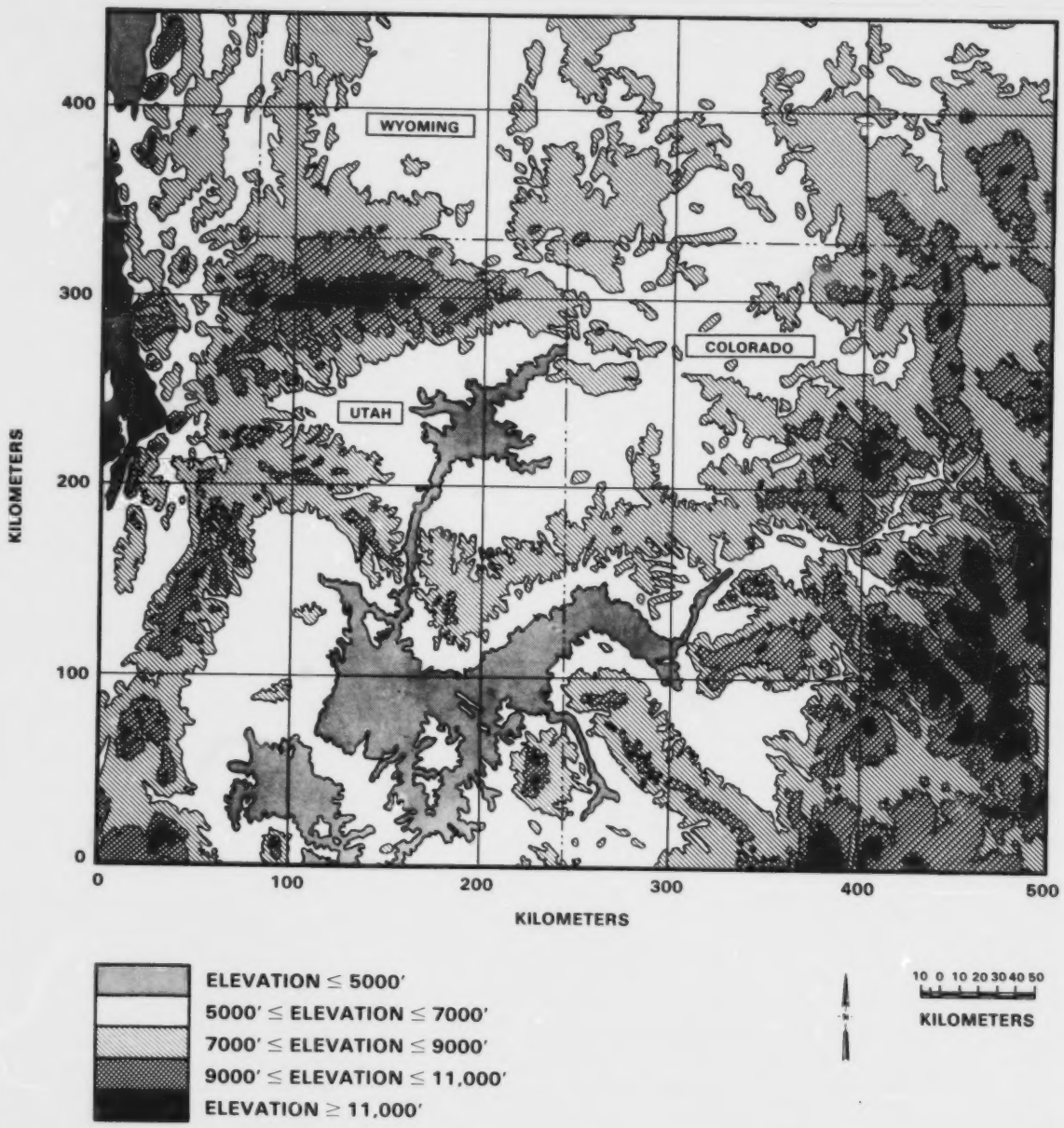


FIGURE A-4

MAJOR RIVERS AND STREAMS IN THE GRAMA MODELING REGION

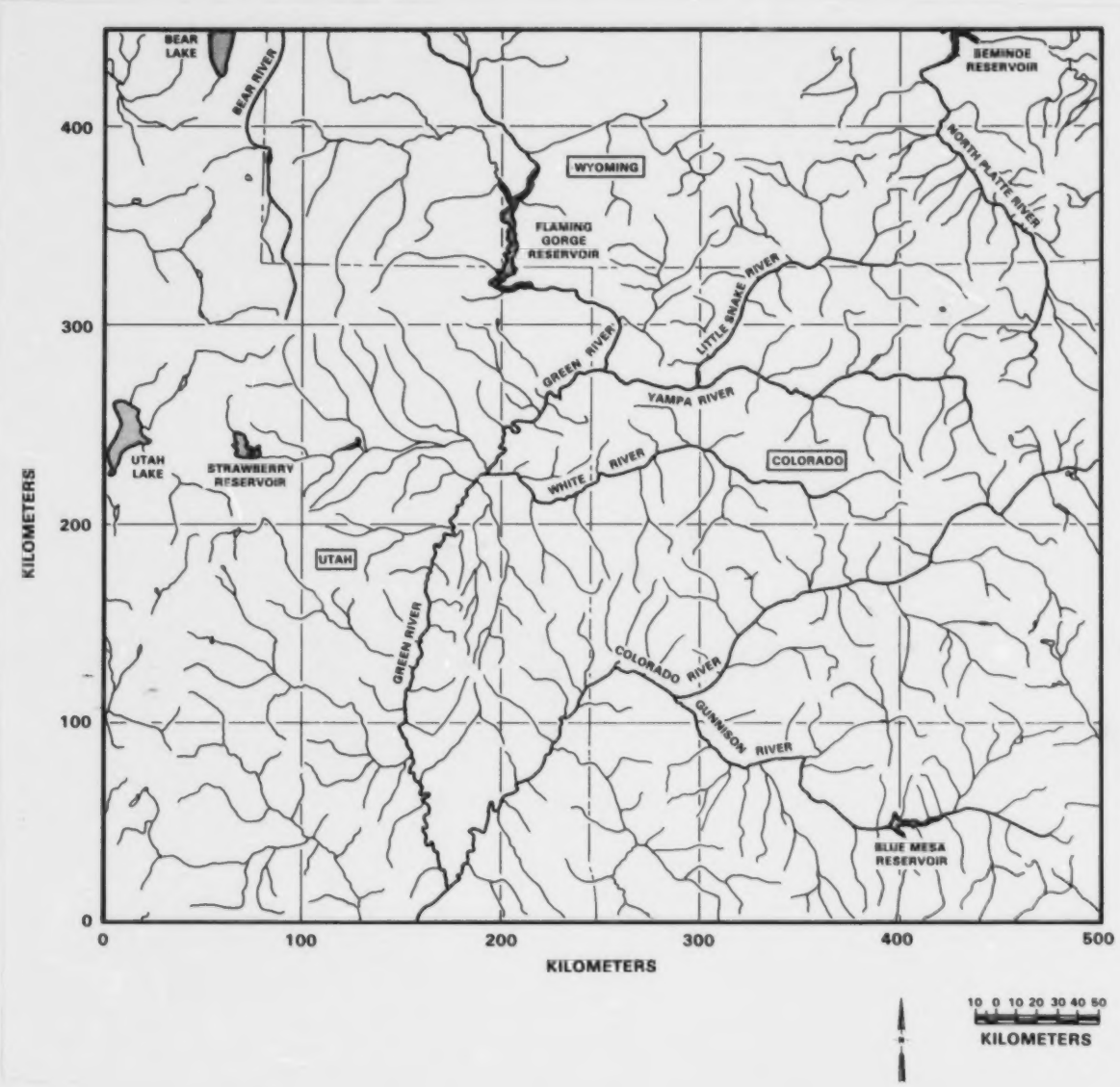
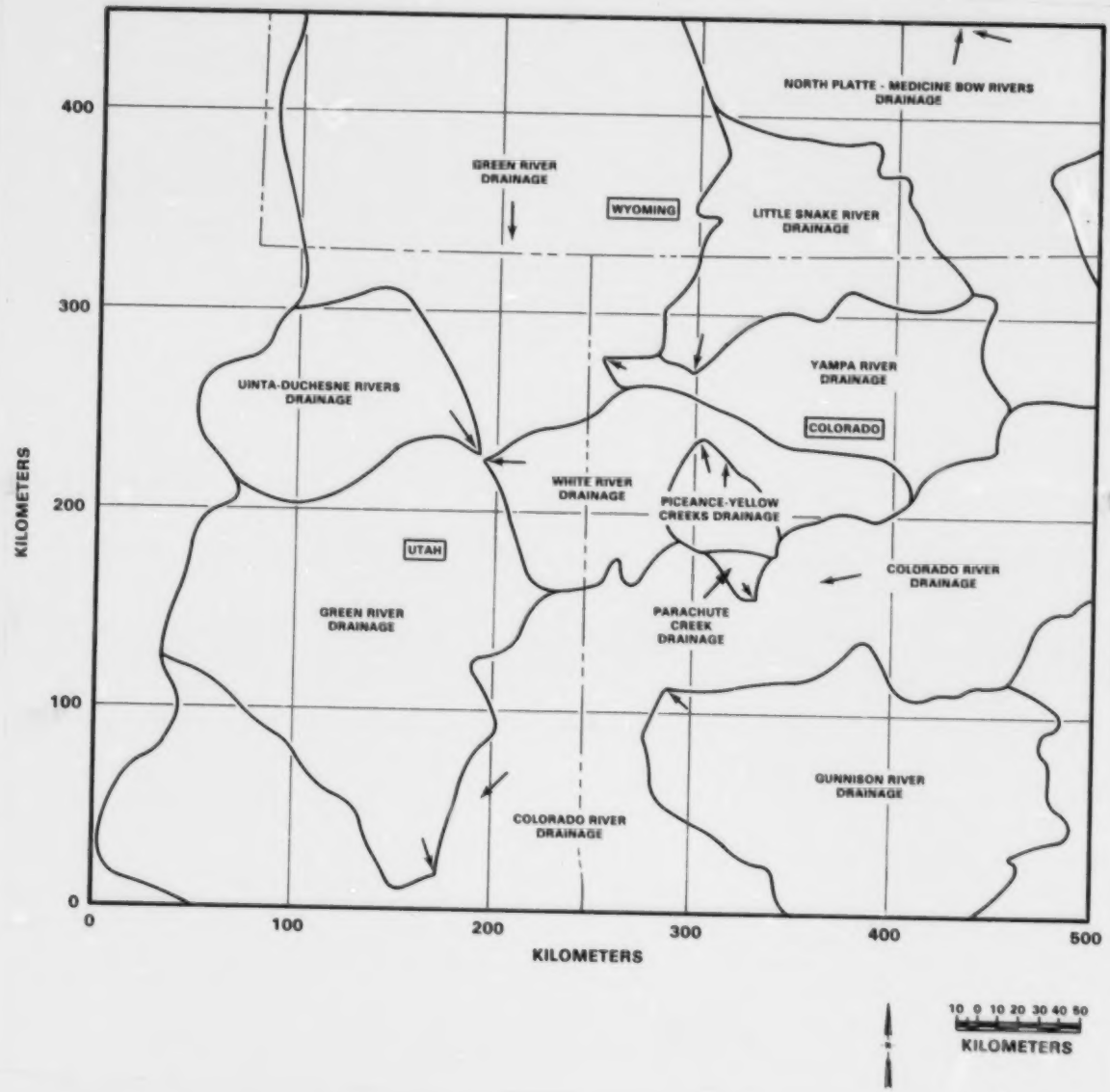


FIGURE A-5

MAIN RIVER DRAINAGES IN THE GRAMA MODELING REGION



APPENDIX B

UNIVAC JCL FILES FOR COMPILING AND MAPPING TERRAIN, MET, POLUT, AND POLPRC

TERRAIN JCL FILE

```
@ FTN,O TERRAIN.AATER
@ EOF
@ MAP,E ,TERRAIN.AATER
IN TERRAIN.AATER
END
```

MET JCL FILE

```
@ FTN,O MET.AAMET
@ EOF
@ MAP,E ,MET.AAMET
IN MET.AAMET
IN RLSEC*IMSL.LLSQF
IN RLSEC*IMSL.SNRM2
IN RLSEC*IMSL.SDOT
IN RLSEC*IMSL.VHS12
IN RLSEC*IMSL.SASUM
IN RLSEC*IMSL.UERTST
IN RLSEC*IMSL.USPKD
IN RLSEC*IMSL.UGETIO
END
```

POLUT JCL FILE

```
@ FTN,O POLUT.AAPOL
@ EOF
@ MAP,E ,POLUT.AAPOL
IN POLUT.AAPOL
END
```

POLPRC JCL FILE

```
@ FTN,O POLPRC.AAPRC
@ EOF
@ MAP,E , POLPRC.AAPRC
IN POLPRC.AAPRC
END
```

APPENDIX C

INSTRUCTIONS AND FORTRAN CODE FOR SETTING UP THE BASE TERRAIN FILE TOPO

The terrain data file, TOPO, used by MELSAR must be a binary file. However, the TOPO file included on the ASCII coded tape submitted with this report is a formatted file. Consequently, the formatted file must be converted to a binary file before it can be used by MELSAR. The FORTRAN program, FORMAT, listed below is used to read the formatted TOPO file and create a binary TOPO file. This program is included on magnetic tape submitted with this report.

```
C   PROGRAM FORMAT.TOPO
C
C       THIS PROGRAM IS USED TO CREATE A TOPOGRAPHICAL BASE BINARY
C    DISK FILE 'TOPO' OF GRAMA DOMAIN FROM FORMATTED DISK FILE.
C    (THE FORMATTED DISK FILE IS FROM CODED STRANGER TAPE.)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        DIMENSION IA (512)
5       DO 10 I=1, 32
        J1 = (I-1)*16+1
        J2 = I*16
10      READ (3,101,END=300)(IA(J),J=J1,J2)
        WRITE (2)(IA(I),I=1,500)
        GO TO 5
101     FORMAT (1615)
300     STOP
        END
```

APPENDIX D

DESCRIPTION OF INPUT DATA FILES AVAILABLE FOR MELSAR

File Name	Date Created	Description
TOPO	April 15, 1985	Base terrain elevation file for GRAMA region. 500 x 450 grid of terrain heights in feet, where each grid element is the average elevation in a 1-km-square area.
UP1	April 15, 1985	Upper-air weather data from six National Weather Service (NWS) radiosonde stations for the month of January 1978. The stations are Grand Junction, CO, Salt Lake City, UT, Lander, WY, Denver, CO, Winslow, AZ, and Albuquerque, NM.
UP2	April 15, 1985	Upper-air weather data for the same stations as in UP1 for the month of July 1978.
SFC1	April 15, 1985	Surface weather data from eight NWS stations for the month of January 1978. The stations are the same as in UP1 with the addition of Eagle, CO, and Rock Springs, WY.
SFC2	April 15, 1985	Surface weather data for the same stations as in SRC1 for the month of July 1978.
WIP1	April 15, 1985	Twenty-two wind interpolation points on the boundary of the GRAMA region. A listing of the file will show the coordinates of the points.
LANDU1	April 15, 1985	Grid of land-use categories over the GRAMA region. The grid is 10 x 9 where each grid cell is 50-km square. The land-use categories were gridded from vegetation types maps of the states of Colorado and Utah. The southern part of Wyoming was assumed to be all desert shrubland.

VALLY1	April 15, 1985	Characteristic for eight valley segments from the head of Clear Creek down Roan Creek and the Colorado River to 30 km below DeBeque, Colorado. An average downvalley wind speed of 4.1 m/s and a maximum potential temperature lapse rate of 0.035 °K/m is assumed.
SOURC1	April 15, 1985	Source inventory of three hypothetical sources used in the example problem. Section 3.0 gives a description of these sources.
RECPT1	April 15, 1985	Three receptor grids and three individual receptors used in the sample problem. Section 3.0 gives a description of these receptors.

(New entries to this list should be made as new data sets are constructed.)

APPENDIX E

BLANK FORMS FOR KEEPING TRACK OF FILES

SUMMARY OF TERRAIN RUNS

RUN #	DOMAIN			

SUMMARY OF MET RUNS

RUN #	DOMAIN	PERIOD		

SUMMARY OF POLUT RUNS

RUN #	DOMAIN	PERIOD	SOURCES	RECEPTORS

SUMMARY OF **POLPRC** RUNS

RUN #	DOMAIN	PERIOD	SOURCES	RECEPTORS

TERRAIN INPUT/OUTPUT FILES

RUN #_____ DATE _____

RUN SPECIFICATION FILE: TERIN__

INPUT FILES	OUTPUT FILES
TOPO	TERF__
	TERS__

MET INPUT/OUTPUT FILES

RUN #_____ DATE _____

RUN SPECIFICATION FILE: METIN__

INPUT FILES	OUTPUT FILES
UP__	FIGI__
SFC__	BLPAR__
WIP__	MIXST__
LANDU__	COEFF__
VALLY__	
TERF__	
TERS__	

POLUT INPUT/OUTPUT FILES

RUN #_____ DATE _____

RUN SPECIFICATION FILE: POLIN__

INPUT FILES	OUTPUT FILES
SOURC__	CONC__
RECPT__	
TERS__	
FIGI__	
BLPAR__	
MIXST__	
COEFF__	

POLPRC INPUT/OUTPUT FILES

RUN #_____ DATE _____

RUN SPECIFICATION FILE: PRCIN__

INPUT FILES	OUTPUT FILES
CONC__	Concentration Tables

APPENDIX F
FILE TRACKING FORMS FOR SAMPLE PROBLEM

SUMMARY OF **TERRAIN** RUNS

RUN #	DOMAIN			
04158512	Whole GRAMA Region			

FIGURE F-1
SUMMARY OF TERRAIN RUNS FOR SAMPLE PROBLEM

FIGURE F-2

SUMMARY OF MET RUNS FOR SAMPLE PROBLEM

SUMMARY OF MET RUNS

RUN #	DOMAIN	PERIOD		
04158512	Whole GRAMA Region	07-11-78 0700 to 07-13-78 0700		

FIGURE F-3

SUMMARY OF POLUT RUNS FOR SAMPLE PROBLEM

SUMMARY OF **POLUT** RUNS

RUN #	DOMAIN	PERIOD	SOURCES	RECEPTORS
04158512	Whole GRAMA Region	07-11-78 0700 to 07-13-78 0700	3 Point Sources (1 Valley)	2 Grids 3 Individual

FIGURE F-4

SUMMARY OF POLPRC RUNS FOR SAMPLE PROBLEM

SUMMARY OF **POLPRC** RUNS

RUN #	DOMAIN	PERIOD	SOURCES	RECEPTORS
04158512	Whole GRAMA Region	07-11-78 0700 to 07-13-78 0700	3 Sources (1 Valley)	2 Grids 3 Individual

FIGURE F-5

TERRAIN INPUT/OUTPUT FILES FOR SAMPLE PROBLEM

TERRAIN INPUT/OUTPUT FILES

RUN # 04158512 DATE April 15, 1985

RUN SPECIFICATION FILE: TERIN.1

INPUT FILES	OUTPUT FILES
TOPO	TERF.1
	TERS.1

FIGURE F-6

MET INPUT/OUTPUT FILES FOR SAMPLE PROBLEM

MET INPUT/OUTPUT FILES

RUN # 04158512 DATE April 15, 1985

RUN SPECIFICATION FILE: METIN 1

INPUT FILES	OUTPUT FILES
UP 2	FIGI 1
SFC 2	BLPAR 1
WIP 1	MIXST 1
LANDU 1	COEFF 1
VALLY 1	
TERF 1	
TERS 1	

FIGURE F-7

POLUT INPUT/OUTPUT FILES FOR SAMPLE PROBLEM

POLUT INPUT/OUTPUT FILES

RUN # 04158512 DATE April 15, 1985

RUN SPECIFICATION FILE: POLIN 1

INPUT FILES	OUTPUT FILES
SOURC 1	CONC 1
RECPT 1	
TERS 1	
FIGI 1	
BLPAR 1	
MIXST 1	
COEFF 1	

FIGURE F-8

POLPRC INPUT/OUTPUT FILES FOR SAMPLE PROBLEM

POLPRC INPUT/OUTPUT FILES

RUN # 04158512 DATE April 15, 1985

RUN SPECIFICATION FILE: PRCIN1

INPUT FILES	OUTPUT FILES
CONC1	Concentration Tables

APPENDIX G

DIAGNOSTIC MESSAGES FOR MET, POLUT, AND POLPRC

DIAGNOSTIC MESSAGES FOR MET

Program AAMET

1) DOMAIN SPECS FROM TERRAIN FILE NOT SAME AS RUN.

   Be sure terrain file used is the file required for this run.

Subroutine AMPLIT

2) UPPER AIR STATION ID AND SURFACE STATION ID DO NOT MATCH.

   Check file UP_ and file SFC_ to be sure no two stations have the same identification number.

Subroutine BLPARS

3) AMPLITUDE FILE CLOCK DOES NOT MATCH BLPARS CLOCK.

   Possible error in code - code needs checked.

4) NO SURFACE STATIONS FOR HOUR ___ ON DAY ___.

   Need to add surface weather data to file SFC_ for period shown, or use a different file.

Subroutine COEFF

5) VALLEY SEGMENTS INFO BEING READ IS NOT FOR VALLEY ID___ DESIRED.

   Check to see that file VALLY_ is constructed properly.

6) NO SOUNDINGS FOR VALLEY ___ ON DAY ___.

   Need to add upper-air data near valley to file UP_ for period given.

7) INVERSION BREAKUP FOR VALLEY ___ WENT PAST SUNSET ON DAY ___.

Possible error in code - code needs checked.

Subroutine FILLMB

8) MISSING PRESSURE NOT FOUND IN 10 ITERATIONS.

Possible error in code - code needs checked.

Subroutine FINDU

9) ID OF SOUNDING DOES NOT CORRESPOND WITH ID REQUIRED.

Possible error in code - code needs checked.

Subroutine FROUDE

10) DOMAIN SPECS IN FROUND # MODIFICATION DO NOT MATCH.

Be sure Froude terrain file used is the file required for this run.

Subroutine HPROF

11) NEED MORE THAN ONE OBSERVATION IN PROFILE.

Check printout summary for subroutine UPPER to be sure all profiles have more than one observation.

Subroutine JULIAN

12) MONTH INDEX IS OUT OF RANGE.

Check to be sure IBEG and IEND in file METIN_ are specified properly.

Subroutine LAPSE

13) NO STATIONS FOR COMPUTATION OF DTDZ AND AVE TEMP AT HOUR ___ ON JULIAN DAY ___.

Check printout summary for subroutine UPPER to be sure upper-air data are available at specified time across grid within at least 500-km increments. If not, include new data in file UP_, or use different UP_ file.

Subroutine MIXSTB

14) NO STATIONS AVAILABLE FOR COMPUTING MIXING HEIGHTS AND STABILITIES.

Do not have upper-air and surface data at the same location for any stations. Include new data in UP_ and SFC_ files used or use different files.

15) ID OF SURFACE DATA DOES NOT CORRESPOND WITH ID REQUIRED.

Possible error in code - code needs checked.

16) MECHANICAL MIXING HEIGHT GREATER THAN HIGHEST OBSERVATION.

Possible error in code - code needs checked. Or check summary printout for subroutine UPPER to be sure the top observation in all soundings is greater than about 1000 m above ground level.

17) NO STATIONS FOR COMPUTATION OF MIXING HT. AND STB ON JULIAN DAY ___.

Check files UP_ and SFC_ to be sure data are available at required time. If not, add new data or use different input files.

18) AMPLITUDE FILE CLOCK DOES NOT MATCH MIXSTB CLOCK.

Possible error in code - code needs checked.

Subroutine SRFC

19) STATION IDS DO NOT MATCH IN SRFC.

Check that file SFC_ is formatted properly and that the number of surface observations specified for each station matches the number of observations included.

Subroutine SUBAO

20) NO STATIONS TO COMPUTE AO ON JULIAN DAY ___.

Check files UP_ and SFC_ to be sure data are available at required time. If not, add new data or use different input files.

Subroutine TEMP

21) ID OF SURFACE DATA DOES NOT CORRESPOND WITH ID REQUIRED.

Possible error in code - code needs checked.

22) MECHANICAL MIXING HEIGHT GREATER THAN HIGHEST OBSERVATION.

Possible error in code - code needs checked. Or check summary printout for subroutine UPPER to be sure the top observation in all soundings is greater than about 1000 m above ground level.

23) NO STATIONS AVAILABLE ON TO COMPUTE POTENTIAL TEMP. LAPSE RATE AND BULK TEMP.

Do not have upper-air and surface data at the same location for any stations. Include new data in UP_ and SFC_ files used or use different files.

Subroutine THETA

24) THETA DID NOT CONVERGE IN 10 ITERATIONS.

Possible error in code - code needs checked.

Subroutine TPROF

25) NEED MORE THAN ONE OBSERVATION IN PROFILE.

Check printout summary for subroutine UPPER to be sure all profiles have more than one observation.

Subroutine UPPER

26) STATION TO BE PROCESSED NOT SAME AS SOUNDING READ.

Check that file UP_ is formatted properly and that the number of soundings specified for each station matches the number of soundings included.

Subroutine WINDS

27) THE GAMMA COORDINATE IS GREATER THAN 1.0 IN SUBROUTINE WINDS.

Possible error in code - code needs checked.

28) GAMMA OF POINT NOT WITHIN GAMMA DOMAIN IN SUBROUTINE WINDS.

Possible error in code - code needs checked.

DIAGNOSTIC MESSAGES FOR POLUT

Program AAPOL

1) SIMULATION END TIME AFTER MET DATA SET END TIME.

Must restart POLUT with IEND before the MET ending time.

Subroutine CLEAN

2) PUFF ARRAYS ARE FULL--MUST RUN POLUT AGAIN FOR LESS TIME OR LOWER PUFF RELEASE RATE.

Subroutine CLEANV

3) PUFF ARRAYS ARE FULL--MUST RUN POLUT AGAIN FOR LESS TIME OR LOWER PUFF RELEASE RATE.

Subroutine DIVIDE

4) SEGMENT LENGTH LESS THAN OR EQUAL TO ZERO.

Possible error in code--code needs checked.

Subroutine INIT

5) SOURCE ____ IS NOT IN MODELING DOMAIN.

Restart POLUT with sources specified in modeling domain.

6) DOMAIN SPECS SPECIFIED IN TERRAIN FILE NOT SAME AS SPECIFIED IN RUN.

Be sure domain specifications in file POLIN_ are compatible with input terrain file TERS_.

7) DOMAIN SPECS SPECIFIED IN WINDS FILE NOT SAME AS SPECIFIED IN RUN.

Be sure domain specifications in POLIN_ are compatible with input wind file FIGI_.

8) DOMAIN SPECS FROM STABILITY FILE NOT SAME AS SPECIFIED IN RUN.

Be sure domain specifications in POLIN_ are compatible with input MET file MIXST_.

9) DOMAIN SPECS FROM MIXING HEIGHT FILE NOT SAME AS SPECIFIED IN RUN.

Be sure domain specifications in POLIN_ are compatible with input MET file MIXST_.

10) REQUESTED SIMULATION START TIME IS NOT INCLUDED IN MET DATA PERIOD.

Restart POLUT with simulation start time within period in MET data files.

11) VALLEY SPECS FOR SOURCE ____ NOT INCLUDED IN COEFF INPUT FILE.

Must specify an input COEFF_ file (from MET) which contains valley of interest.

Subroutine WINDS

12) GAMMA OF POINT NOT WITHIN GAMMA DOMAIN IN SUBROUTINE WINDS.

Possible error in code--code needs checked.

Subroutine YVIRT

13) VIRT. DIST. COMP. DID NOT CONVERGE IN 20 ITER.

Possible error in code--code needs checked.

Subroutine YVIRTG

14) VIRT. DIST. COMP. DID NOT CONVERGE IN 20 ITER.

Possible error in code--code needs checked.

DIAGNOSTIC MESSAGES FOR POLPRC

Program AAPRC

1) PUFF RELEASE RATE FOR FILE ____ DOES NOT MATCH OTHER INPUT FILES.

Be sure puff release rate in the input CONC_ files are the same.

2) TIME PERIOD OF FILE ____ DOES NOT MATCH PERIOD REQUESTED.

Be sure time periods in the input CONC_ files are the same.

3) DOMAIN OF FILE ____ DOES NOT MATCH PERIOD SPECIFIED.

Be sure the domain size in the input CONC_ files are the same.

4) NUMBER OF RECEPTORS FOR FILE ____ DOES NOT MATCH OTHER INPUT FILES.

Be sure receptors in the input CONC_ files are the same.

5) RECEPTOR GRIDS OF FILE ____ DO NOT MATCH THOSE IN OTHER INPUT FILES.

Be sure receptors in the input CONC_ files are the same.

6) IND RECEPTORS FOR FILE ____ DO NOT MATCH THOSE IN OTHER INPUT FILES.

Be sure receptors in the input CONC_ files are the same.

7) POLLUTANTS FOR FILE ____ DO NOT MATCH IN OTHER INPUT FILES.

Be sure pollutant types in the input CONC_ files are the same.

8) GREATER THAN 20 SOURCES IN INPUT FILES.

Be sure number of sources of all CONC_ input files does not total more than twenty.

Subroutine JULIAN

9) MONTH INDEX IS OUT OF RANGE.

Check to be sure IBEG and IEND in file PRCIN_ are specified properly.

APPENDIX H

SUMMARY OUTPUT LISTING FROM TERRAIN FOR SAMPLE PROBLEM

```
****SUMMARY RESULTS OF TERRAIN - TERRAIN PROCESSING PROGRAM****

RUN IDENTIFICATION:  04158512   (FROUDE GRID DATA)


**DOMAIN**

  REGION SIZE:   500.0  BY    450.0  KILOMETERS
  GRAMA COORDINATES (KM):   X=      0.0  TO    500.0
                            Y=      0.0  TO    450.0


**OUTPUT TERRAIN DATA**

  AVERAGE TERRAIN STATISTICS WERE COMPUTED FROM THE BASE TERRAIN
  FILE USING THE FOLLOWING GRID CELL SPECIFICATIONS:
       CELL SIZE:   50 BY   50 KILOMETERS
       NUMBER OF CELLS IN X-DIR:   10
       NUMBER OF CELLS IN Y-DIR:    9
       SMOOTHING INTERVAL:   50 KILOMETERS
```

```
****SUMMARY RESULTS OF TERRAIN - TERRAIN PROCESSING PROGRAM****

RUN IDENTIFICATION:  04158512   (SMOOTH TERRAIN GRID DATA)

**DOMAIN**

  REGION SIZE:   500.0  BY    450.0  KILOMETERS
  GRAMA COORDINATES (KM):   X=     0.0  TO    500.0
                            Y=     0.0  TO    450.0

**OUTPUT TERRAIN DATA**

  AVERAGE TERRAIN STATISTICS WERE COMPUTED FROM THE BASE TERRAIN
  FILE USING THE FOLLOWING GRID CELL SPECIFICATIONS:
       CELL SIZE:   10 BY   10 KILOMETERS
       NUMBER OF CELLS IN X-DIR:   50
       NUMBER OF CELLS IN Y-DIR:   45
       SMOOTHING INTERVAL:   20 KILOMETERS
```

APPENDIX I

SUMMARY OUTPUT LISTING FROM MET FOR SAMPLE PROBLEM

```
*****************************************************************************************
*                                                                                       *
*                                                                                       *
*                                    PROGRAM MET                                        *
*                          +++METEOROLOGICAL DATA PROCESSOR+++                          *
*   PROVIDES HOURLY GRIDDED FIELDS OF MIXING HEIGHTS, STABILITIES, TEMPERATURES, PRESSURES.  *
*   FRICTION VELOCITIES, MONIN-OBUKOV LENGTHS, AND CONVECTIVE VELOCITIES.  HOURLY 3-D      *
*   ANALYTICAL WIND FIELDS ARE PRODUCED IN ADDITION TO HOURLY VALLEY COUPLING COEFFICIENTS.  *
*                                                                                       *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*
*                                                                                       *
*     THE RESULTS OF RUN  04158512  ARE GIVEN.                    USER IS  MR JONES     *
*     PROCESSING START TIME IS (MODAYRHR) 07117807     [ JULIAN DAY IS 192  HOUR IS  7 ]  *
*     PROCESSING  STOP TIME IS (MODAYRHR) 07137807     [ JULIAN DAY IS 194  HOUR IS  7 ]  *
*                                                                                       *
*****************************************************************************************


THE RUN SPECIFICATIONS ARE (FILE METIN):

 IDRUN =  04158512   IBEG = 07117807    IEND = 07137807    IBTZ =  7          USENAM =  MR JONES

 DETAIL =  F         NSTA =   6         NSSTA =   8        NVALE =   1        NOSTA =  22

 IXO =   0           IYO =   0          SLX =  500.        SLY =  450.

 ALAT =  40.000      URLONG = 109.000

 NTEMAX =  10        IHH =  1           HH =  1500.        NGAMA =  5         FROU =  F

 RAD =  500.         ZNOT = 999.00      DS =    1.

 IXOS =   0          IYOS =   0         NXCS = 10          NYCS =  9          ICSS =  50

 HTGAMA =     0.0033    0.0670    0.2000    0.5330    1.0000


THE TERRAIN GRID SPECIFICATIONS ARE (FROM TERRAIN FILE):

 IXOT =   0          IYOT =   0         NXCT = 50          NYCT = 45          ICST =  10
```

TRACE OF SUBROUTINE UPPER EXECUTION - UPPER READS IN ALL UPPER AIR DATA FROM THE INPUT FILE AND WRITES OUT DIRECT ACCESS DISK FILES FOR USE BY OTHER PROGRAM COMPONENTS. THE BASE TIME ZONE IS MST .

SUMMARY OF STATIONS PROCESSED:

NAME	ID NO	X (KM)	Y (KM)	LAT (DEG)	LONG (DEG)	ELEV (M MSL)	NO TMP SND	NO WND SND
ALBUQUERQU	23050	467.2	-328.5	35.050	106.617	1620.0	62	62
DENVER	23062	603.5	199.2	39.750	104.867	1625.0	62	62
GRAND JCT.	23066	290.8	120.8	39.117	108.533	1475.0	62	62
WINSLOW	23194	92.8	-333.4	35.017	110.733	1488.0	61	61
LANDER	24021	272.3	531.8	42.817	108.733	1694.0	62	62
SALT LAKE	24127	0.9	308.2	40.767	111.967	1288.0	62	62

THE WINDS FOR EACH STATION ARE OUTPUT AT THE FOLLOWING HEIGHTS (M MSL) CORRESPONDING TO THE IDENTIFIED "GAMMA SURFACES" :

NAME	"0.0033"	"0.0670"	"0.2000"	"0.5330"	"1.0000"	"
ALBUQUERQU	1624.9	1720.5	1920.0	2419.5	3120.0	
DENVER	1629.9	1725.5	1925.0	2424.5	3125.0	
GRAND JCT.	1479.9	1575.5	1775.0	2274.5	2975.0	
WINSLOW	1492.9	1588.5	1788.0	2287.5	2988.0	
LANDER	1698.9	1794.5	1994.0	2493.5	3194.0	
SALT LAKE	1292.9	1388.5	1588.0	2087.5	2788.0	

TRACE OF SUBROUTINE SRFC EXECUTION - SRFC READS IN ALL SURFACE DATA FROM THE INPUT FILE AND WRITES OUT DIRECT ACCESS DISK FILES FOR USE BY OTHER PROGRAM COMPONENTS. THE BASE TIME ZONE IS MST .

SUMMARY OF STATIONS PROCESSED:

NAME	ID NO	X (KM)	Y (KM)	LAT (DEG)	LONG (DEG)	ELEV (M MSL)	NO OBS	BEG DA	BEG HR	END DA	END HR
ALBUQUERQU	23050	467.2	-328.5	35.050	106.617	1620.0	744	181	24	212	23
DENVER	23062	603.5	199.2	39.750	104.867	1625.0	248	182	20	212	17
EAGLE	23063	428.7	182.0	39.650	106.917	1985.0	248	182	2	212	23
GRAND JCT.	23066	290.8	120.8	39.117	108.533	1475.0	744	181	24	212	23
WINSLOW	23194	92.8	-333.4	35.017	110.733	1488.0	248	182	2	212	23
ROCK SPRGS	24027	244.9	396.5	41.600	109.067	2056.0	237	182	2	212	23
SALT LAKE	24127	0.9	308.2	40.767	111.967	1288.0	744	181	24	212	23
LANDER	24021	272.3	531.8	42.817	108.733	1694.0	744	181	24	212	23

TRACE OF SUBROUTINE AMPLIT EXECUTION - AMPLIT COMPUTES THE AMPLITUDE FUNCTIONS FOR EACH HOUR FOR EACH "GAMMA" SURFACE USING THE HOURLY INPUT WINDS FOR EACH "GAMMA" SURFACE. THE BASE TIME ZONE IS MST .

THE WINDS ARE COMPUTED ON THE FOLLOWING "GAMMA" SURFACES :
GAMMA = 0.0033 0.0670 0.2000 0.5330 1.0000
THE UPPER BOUNDARY IS AT A CONSTANT 1500. M AGL.
THE NUMBER OF POLYNOMIAL TERMS ARE 10 .

SUMMARY OF OTHER SPECIFIED POINTS WHERE THE INPUT WINDS ARE INTERPOLATED TO:

NAME	ID NO	X (KM)	Y (KM)
SW CORNER	18	0.0	0.0
	19	0.0	75.0
	20	0.0	150.0
	21	0.0	225.0
	22	0.0	300.0
	23	0.0	375.0
NW CORNER	24	0.0	450.0
SE CORNER	25	500.0	0.0
	26	500.0	75.0
	27	500.0	150.0
	28	500.0	225.0
	29	500.0	300.0
	30	500.0	375.0
NE CORNER	31	500.0	450.0
	32	100.0	0.0
	33	200.0	0.0
	34	300.0	0.0
	35	400.0	0.0
	36	100.0	450.0
	37	200.0	450.0
	38	300.0	450.0
	39	400.0	450.0

TRACE OF SUBROUTINE MIXSTB EXECUTION - MIXSTB COMPUTES HOURLY GRIDDED MIXING HEIGHTS AND STABILITIES FROM UPPER AIR AND SURFACE TEMPERATURE OBSERVATIONS AND WINDS FROM THE FLOW MODEL. THE BASE TIME ZONE IS MST .

SUMMARY OF STATIONS USED:

NAME	ID NO	X (KM)	Y (KM)	LAT (DEG)	LONG (DEG)	ELEV (M MSL)
ALBUQUERQU	23050	467.2	-328.5	35.050	106.617	1620.0
DENVER	23062	603.5	199.2	39.750	104.867	1625.0
GRAND JCT.	23066	290.8	120.8	39.117	108.533	1475.0
WINSLOW	23194	92.8	-333.4	35.017	110.733	1488.0
LANDER	24021	272.3	531.8	42.817	108.733	1694.0
SALT LAKE	24127	0.9	308.2	40.767	111.967	1288.0

THE GRID SPECIFICATIONS ARE:

IXOS = 0 IYOS = 0 NXCS = 10 NYCS = 9 ICSS = 50

TRACE OF SUBROUTINE BLPARS EXECUTION - BLPARS COMPUTES HOURLY GRIDDED BOUNDARY LAYER PARAMETERS FROM SURFACE WEATHER OBSERVATIONS. THE BASE TIME ZONE IS MST .

THE GRID SPECIFICATIONS ARE:
IXOS = 0 IYOS = 0 NXCS = 10 NYCS = 9 ICSS = 50

THE INPUT SURFACE ROUGHNESS IS (M):

	X (KM)									
Y (KM)	25.	75.	125.	175.	225.	275.	325.	375.	425.	475
425.	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
375.	1.00	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
325.	1.00	1.00	1.00	1.00	0.30	0.30	0.30	0.30	0.30	0.30
275.	1.00	1.00	1.00	0.30	0.30	0.30	0.30	0.30	1.00	0.30
225.	1.00	1.00	0.30	0.30	0.30	0.30	0.30	0.90	0.30	0.30
175.	0.30	1.00	0.30	0.30	0.30	0.90	0.90	0.30	1.00	0.30
125.	1.00	0.30	0.30	1.00	0.30	0.30	0.30	0.30	0.30	0.30
75.	1.00	0.30	0.30	0.20	0.30	0.30	0.30	0.90	0.30	0.30
25.	0.30	0.20	0.20	0.30	0.30	0.30	0.90	0.90	0.30	0.30

TRACE OF SUBROUTINE COEFF EXECUTION - COEFF COMPUTES THE COUPLING COEFFICIENT FOR ALL VALLEY SEGMENTS FOR EACH PROCESSING TIME STEP OF 3600. SEC. THE BASE TIME ZONE IS MST .

THE VALLEYS PROCESSED ARE:

NAME	ID NO	X (KM)	Y (KM)	LAT (DEG)	LONG (DEG)	ELEV (M MSL)	DEPTH (M)	WS (M/S)	GAM (DEGK/M)	# SLGS
CLEAR CRK	1	313.5	152.9	39.405	108.265	1680.0	720.	4.1	.035	8

THE SEGMENT DATA FOR VALLEY " CLEAR CRK" ARE:

ID SEG	X (KM)	Y (KM)	EL (M MSL)	TOP EL (M MSL)	WIDTH (M)	ANG1 (DEG)	ANG2 (DEG)
1	313.0	129.5	1460.0	2290.0	1800.	9.	18.
2	317.3	142.6	1520.0	2290.0	5200.	13.	16.
3	315.2	148.7	1550.0	2290.0	1800.	16.	20.
4	311.1	157.0	1620.0	2440.0	1500.	35.	35.
5	308.4	160.8	1650.0	2440.0	1500.	34.	29.
6	309.7	165.2	1710.0	2470.0	750.	34.	34.
7	305.5	172.6	1920.0	2470.0	500.	29.	38.
8	301.1	176.7	1980.0	2500.0	250.	27.	40.

APPENDIX J
SUMMARY OUTPUT LISTING FROM POLUT FOR SAMPLE PROBLEM

```
****SUMMARY RESULTS OF POLUT - POLLUTANT CONCENTRATION MODEL****

  USER:  MR JONES
  RUN IDENTIFICATION:  04158512
  BEGINNING TIME (MMDDYYHH): 07117807  ENDING TIME: 07137807
         (JULIAN DAY,HOUR): (192, 7)                (194, 7)

**RUN SPECIFICATIONS**

  THE PUFF RELEASE RATE WAS    1 PUFFS PER HOUR.
  UNIFORM VERTICAL MIXING WAS NOT ASSUMED.
  THE VERTICAL VELOCITY WAS NOT USED TO TRANSPORT PUFFS.
  THE METHOD USED TO COMPUTE SIGMA-V AND SIGMA-W WAS:
                IRWINS GENERAL FORMULATION.

**DOMAIN**

  REGION SIZE:   500.0  BY    450.0  KILOMETERS
  GRAMA COORDINATES (KM):   X=     0.0  TO    500.0
                            Y=     0.0  TO    450.0

**INPUT TERRAIN DATA**

  AVERAGE TERRAIN STATISTICS USED AS INPUT WERE COMPUTED FROM THE
  BASE TERRAIN FILE USING THE FOLLOWING GRID CELL SPECIFICATIONS:
       CELL SIZE:   10 BY   10 KILOMETERS
       NUMBER OF CELLS IN X-DIR:   50
       NUMBER OF CELLS IN Y-DIR:   45

**INPUT MET DATA**

  MET DATA SPECIFIED ON FOLLOWING GRID:
  REGION SIZE:   500.0  BY    450.0  KILOMETERS
       NUMBER OF CELLS IN X-DIR:   10
       NUMBER OF CELLS IN Y-DIR:    9

**INPUT SOURCE DATA**

  NUMBER OF POINT SOURCES:      3
    NAME      X-COR   Y-COR   ST HT  EX VE  EX TE   RAD
              (KM)    (KM)    (M)    (M/S)   (K)    (M)
  SOURCE 1   230.0   229.0     60.   0.000   349.   3.00
  SOURCE 2   292.0   210.0     60.   0.000   349.   3.00
  SOURCE 3   300.0   180.0     60.   0.000   349.   3.00
```

SOURCE VALLEYS

NUMBER OF SOURCES LOCATED IN VALLEYS: 1

SOURCE NAME	VALLEY NAME	X-COR (KM)	Y-COR (KM)
SOURCE 3	CLEAR CRK	313.5	152.9

INPUT EMISSIONS DATA

NUMBER OF POLLUTANTS: 1

SOURCE NAME	SO2 (G/S)
SOURCE 1	1000.
SOURCE 2	1000.
SOURCE 3	1000.

RECEPTOR LOCATIONS

NUMBER OF RECEPTOR GRIDS: 2

GRID	NAME	X - DIRECTION STA X (KM)	STP X (KM)	SPACE (KM)	Y - DIRECTION STA Y (KM)	STP Y (KM)	SPACE (KM)
1	FLAT TOPS	375.0	395.0	4.0	200.0	240.0	4.0
2	DINOSAUR	246.0	266.0	2.0	266.0	270.0	2.0

NUMBER OF INDIVIDUAL RECEPTORS: 3

RECEPTOR	NAME	X-CORD (KM)	Y-CORD (KM)
1	SO FLT TOP	385.0	195.0
2	MEEKER	343.0	223.0
3	DINOSAUR	227.0	262.0

APPENDIX K
SUMMARY OUTPUT LISTING FROM POLPRC FOR SAMPLE PROBLEM

```
***** SUMMARY OF POLPRC - CONCENTRATION FIELDS POST PROCESSOR *****

  USER:  MR JONES
  RUN ID:  04158512
  BEGINNING TIME (MMDDYYHH): 07117807  ENDING TIME: 07137807
          (JULIAN DAY,HOUR): (192. 7)               (194. 7)

*** IDENTIFICATION OF INPUT CONCENTRATION FILES ***

   04158512

*** DOMAIN ***

  REGION SIZE: 500.0 BY  450.0  KILOMETERS
  GRAMA COORDINATES (KM):   X=   0.0 TO  500.0
                            Y=   0.0 TO  450.0

*** TIME PERIOD OF SIMULATION ***

  TOTAL HOURS OF SIMULATION:                    48
  NUMBER OF TIME STEPS PER HOUR:                 1
  TOTAL NUMBER OF TIME STEPS IN SIMULATION:     48

*** MOVING AVERAGE PERIODS ***

  MOVING AVERAGING PERIODS COMPUTED:  3-HOURLY

  NUMBER OF AVERAGES FOR  3-HOURLY:   46
```

*** POLLUTANT SOURCES ***

NUMBER OF SOURCES: 3

NAME	X-COR (KM)	Y-COR (KM)
SOURCE 1	230.0	229.0
SOURCE 2	292.0	210.0
SOURCE 3	300.0	180.0

*** RECEPTOR LOCATIONS ***

NUMBER OF RECEPTOR GRIDS: 2

GRID	NAME	X - DIRECTION STA X (KM)	STP X (KM)	SPACE (KM)	Y - DIRECTION STA Y (KM)	STP Y (KM)	SPACE (KM)
1	FLAT TOPS	375.0	395.0	4.0	200.0	240.0	4.0
2	DINOSAUR	246.0	266.0	2.0	266.0	270.0	2.0

NUMBER OF INDIVIDUAL RECEPTORS: 3

RECEPTOR	NAME	X-CORD (KM)	Y-CORD (KM)
1	SO FLT TOP	385.0	195.0
2	MEEKER	343.0	223.0
3	DINOSAUR	227.0	262.0

APPENDIX L

OUTPUT CONCENTRATION TABLES FROM POLPRC FOR SAMPLE PROBLEM

HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: ALL POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0 --
						<< X(Km)
240.0	5.248+0	4.920+0	4.601+0	4.287+0	4.010+0	3.731+0
236.0	5.299+0	4.940+0	4.618+0	4.291+0	3.996+0	3.714+0
232.0	5.355+0	4.976+0	4.637+0	4.303+0	3.979+0	3.693+0
228.0	5.414+0	5.026+0	4.656+0	4.314+0	3.971+0	3.669+0
224.0	5.461+0	5.062+0	4.661+0	4.311+0	3.962+0	3.634+0
220.0	5.474+0	5.065+0	4.650+0	4.276+0	3.923+0	3.581+0
216.0	5.487+0	5.066+0	4.642+0	4.240+0	3.885+0	3.542+0
212.0	5.497+0	5.065+0	4.633+0	4.212+0	3.849+0	3.505+0
208.0	5.503+0	5.060+0	4.622+0	4.197+0	3.814+0	3.469+0
204.0	5.506+0	5.053+0	4.609+0	4.179+0	3.779+0	3.431+0
200.0	5.510+0	5.045+0	4.592+0	4.157+0	3.744+0	3.391+0

TIME OF HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (JULIAN DAY, HOUR)
FOR PERIOD: 07117807 (192.07) TO: 07137807 (194.07)
SOURCE: ALL POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	193,22	193,23	193,23	194,00	194,00	194,00
236.0	193,22	193,23	193,23	193,23	194,00	194,00
232.0	193,22	193,22	193,23	193,23	194,00	194,00
228.0	193,22	193,22	193,23	193,23	193,23	194,00
224.0	193,22	193,22	193,23	193,23	193,23	194,00
220.0	193,22	193,22	193,22	193,23	193,23	193,23
216.0	193,22	193,22	193,22	193,23	193,23	193,23
212.0	193,22	193,22	193,22	193,22	193,23	193,23
208.0	193,22	193,22	193,22	193,22	193,23	193,23
204.0	193,22	193,22	193,22	193,22	193,23	193,23
200.0	193,21	193,22	193,22	193,22	193,22	193,23

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 1 POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	6.978-1	7.427-1	6.636-1	6.852-1	6.142-1	5.492-1
236.0	6.511-1	6.917-1	6.192-1	5.540-1	5.711-1	5.114-1
232.0	6.065-1	5.436-1	5.761-1	5.162-1	5.291-1	4.743-1
228.0	5.631-1	5.056-1	5.339-1	4.791-1	4.297-1	4.382-1
224.0	5.189-1	4.667-1	4.913-1	4.416-1	3.968-1	4.024-1
220.0	4.724-1	4.256-1	3.839-1	4.023-1	3.625-1	3.266-1
216.0	4.277-1	3.862-1	3.489-1	3.651-1	3.299-1	2.981-1
212.0	3.848-1	3.487-1	3.161-1	2.872-1	2.993-1	2.713-1
208.0	3.446-1	3.133-1	2.853-1	2.600-1	2.707-1	2.461-1
204.0	3.074-1	2.807-1	2.566-1	2.346-1	2.440-1	2.224-1
200.0	2.256-1	2.509-1	2.299-1	2.108-1	1.932-1	2.001-1

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 2 POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	3.338+0	3.049+0	2.866+0	2.590+0	2.437+0	2.278+0
236.0	3.318+0	3.021+0	2.833+0	2.636+0	2.392+0	2.231+0
232.0	3.287+0	3.048+0	2.790+0	2.589+0	2.338+0	2.174+0
228.0	3.245+0	2.997+0	2.738+0	2.532+0	2.325+0	2.109+0
224.0	3.182+0	2.926+0	2.667+0	2.457+0	2.248+0	2.032+0
220.0	3.085+0	2.823+0	2.565+0	2.355+0	2.148+0	1.950+0
216.0	2.975+0	2.707+0	2.447+0	2.245+0	2.041+0	1.848+0
212.0	2.850+0	2.578+0	2.319+0	2.077+0	1.929+0	1.742+0
208.0	2.708+0	2.436+0	2.182+0	1.947+0	1.812+0	1.632+0
204.0	2.550+0	2.282+0	2.035+0	1.810+0	1.690+0	1.518+0
200.0	2.156+0	2.117+0	1.881+0	1.668+0	1.477+0	1.401+0

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 3 POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	1.213+0	1.128+0	1.072+0	1.011+0	9.588-1	9.036-1
236.0	1.330+0	1.227+0	1.166+0	1.100+0	1.033+0	9.722-1
232.0	1.461+0	1.384+0	1.271+0	1.198+0	1.113+0	1.045+0
228.0	1.606+0	1.523+0	1.384+0	1.303+0	1.217+0	1.122+0
224.0	1.761+0	1.669+0	1.502+0	1.413+0	1.317+0	1.199+0
220.0	1.916+0	1.816+0	1.702+0	1.518+0	1.413+0	1.305+0
216.0	2.084+0	1.972+0	1.846+0	1.629+0	1.514+0	1.396+0
212.0	2.262+0	2.138+0	1.998+0	1.848+0	1.621+0	1.492+0
208.0	2.451+0	2.311+0	2.156+0	1.990+0	1.732+0	1.590+0
204.0	2.648+0	2.490+0	2.317+0	2.135+0	1.845+0	1.691+0
200.0	3.129+0	2.677+0	2.481+0	2.279+0	2.074+0	1.790+0

<< X(Km)

SECOND HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: ALL POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	5.228+0	4.881+0	4.560+0	4.278+0	3.952+0	3.683+0
236.0	5.250+0	4.927+0	4.553+0	4.276+0	3.962+0	3.646+0
232.0	5.282+0	4.965+0	4.589+0	4.266+0	3.968+0	3.636+0
228.0	5.319+0	4.993+0	4.630+0	4.255+0	3.961+0	3.633+0
224.0	5.376+0	5.008+0	4.657+0	4.252+0	3.930+0	3.620+0
220.0	5.413+0	4.989+0	4.632+0	4.240+0	3.871+0	3.574+0
216.0	5.446+0	4.971+0	4.604+0	4.225+0	3.822+0	3.514+0
212.0	5.471+0	4.966+0	4.576+0	4.207+0	3.805+0	3.459+0
208.0	5.488+0	4.977+0	4.550+0	4.175+0	3.788+0	3.405+0
204.0	5.499+0	4.984+0	4.525+0	4.144+0	3.769+0	3.380+0
200.0	5.508+0	4.989+0	4.499+0	4.110+0	3.740+0	3.354+0

<< X(Km)

TIME OF SECOND HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (JULIAN DAY, HOUR)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: ALL POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	193,23	193,22	194,00	193,23	193,23	194,01
236.0	193,23	193,22	194,00	194,00	193,23	194,01
232.0	193,23	193,23	193,22	194,00	193,23	193,23
228.0	193,23	193,23	193,22	194,00	194,00	193,23
224.0	193,21	193,23	193,22	193,22	194,00	193,23
220.0	193,21	193,23	193,23	193,22	194,00	194,00
216.0	193,21	193,23	193,23	193,22	193,22	194,00
212.0	193,21	193,21	193,23	193,23	193,22	194,00
208.0	193,21	193,21	193,23	193,23	193,22	194,00
204.0	193,21	193,21	193,23	193,23	193,22	193,22
200.0	193,22	193,21	193,23	193,23	193,23	193,22

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE SECOND HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 1 POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	8.305-1	6.227-1	7.627-1	5.926-1	5.290-1	6.216-1
236.0	7.719-1	5.825-1	7.073-1	6.363-1	4.954-1	5.746-1
232.0	7.160-1	6.425-1	4.878-1	5.889-1	4.623-1	4.138-1
228.0	6.617-1	5.946-1	4.544-1	5.427-1	4.882-1	3.853-1
224.0	4.259-1	5.463-1	4.202-1	3.784-1	4.475-1	3.565-1
220.0	3.882-1	4.958-1	4.466-1	3.465-1	4.058-1	3.658-1
216.0	3.518-1	4.478-1	4.042-1	3.160-1	2.866-1	3.314-1
212.0	3.167-1	2.881-1	3.644-1	3.302-1	2.612-1	2.994-1
208.0	2.838-1	2.589-1	3.274-1	2.977-1	2.372-1	2.697-1
204.0	2.532-1	2.319-1	2.932-1	2.675-1	2.145-1	1.961-1
200.0	2.739-1	2.072-1	2.617-1	2.394-1	2.190-1	1.770-1

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE SECOND HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 2 POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	3.218+0	3.112+0	2.737+0	2.674+0	2.476+0	2.165+0
236.0	3.196+0	3.085+0	2.701+0	2.549+0	2.436+0	2.115+0
232.0	3.169+0	2.985+0	2.802+0	2.500+0	2.386+0	2.183+0
228.0	3.134+0	2.940+0	2.746+0	2.443+0	2.276+0	2.122+0
224.0	3.075+0	2.877+0	2.670+0	2.419+0	2.200+0	2.046+0
220.0	2.967+0	2.784+0	2.568+0	2.314+0	2.102+0	1.936+0
216.0	2.840+0	2.682+0	2.460+0	2.199+0	1.967+0	1.835+0
212.0	2.694+0	2.585+0	2.343+0	2.129+0	1.852+0	1.731+0
208.0	2.529+0	2.230+0	2.217+0	2.007+0	1.732+0	1.624+0
204.0	2.348+0	2.064+0	2.083+0	1.878+0	1.606+0	1.423+0
200.0	2.377+0	1.889+0	1.942+0	1.743+0	1.563+0	1.306+0

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE SECOND HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 3 POLLUTANT: SO2 GRID: FLAT TOPS

Y(Km)	375.0	379.0	383.0	387.0	391.0	395.0
240.0	1.179+0	1.147+0	1.061+0	1.011+0	9.474-1	8.967-1
236.0	1.282+0	1.259+0	1.145+0	1.091+0	1.030+0	9.563-1
232.0	1.397+0	1.337+0	1.300+0	1.177+0	1.120+0	1.039+0
228.0	1.523+0	1.458+0	1.429+0	1.270+0	1.197+0	1.126+0
224.0	1.875+0	1.585+0	1.566+0	1.455+0	1.283+0	1.217+0
220.0	2.058+0	1.709+0	1.618+0	1.579+0	1.364+0	1.273+0
216.0	2.254+0	1.842+0	1.740+0	1.710+0	1.568+0	1.348+0
212.0	2.461+0	2.292+0	1.869+0	1.747+0	1.692+0	1.428+0
208.0	2.675+0	2.488+0	2.005+0	1.871+0	1.819+0	1.511+0
204.0	2.897+0	2.688+0	2.148+0	1.999+0	1.948+0	1.761+0
200.0	2.858+0	2.894+0	2.296+0	2.128+0	1.959+0	1.871+0

<< X(Km)

HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: ALL POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264 0	266.0	<< X(Km)
270.0	1.523+1	1.504+1	1.512+1	1.623+1	1.796+1	2.066+1	2.437+1	2.851+1	3.280+1	3.684+1	4.021+1	
268.0	1.523+1	1.508+1	1.536+1	1.683+1	1.959+1	2.358+1	2.837+1	3.374+1	3.930+1	4.454+1	4.888+1	
266.0	1.521+1	1.507+1	1.560+1	1.780+1	2.169+1	2.669+1	3.273+1	3.955+1	4.664+1	5.332+1	5.882+1	

TIME OF HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (JULIAN DAY, HOUR)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: ALL POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0
270.0	193,13	193,13	193,13	193,12	193,12	193,11	193,11	193,11	193,11	193,11	193,11
268.0	193,13	193,13	193,12	193,12	193,11	193,11	193,11	193,11	193,11	193,11	193,11
266.0	193,13	193,13	193,12	193,11	193,11	193,11	193,11	193,11	193,11	193,11	193,11

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 1 POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	1.063+1	9.555+0	8.578+0	6.151+0	5.499+0	4.270+0	3.896+0	3.570+0	3.285+0	3.033+0	2.812+0	
268.0	1.045+1	9.367+0	6.609+0	5.879+0	4.495+0	4.111+0	3.772+0	3.474+0	3.211+0	2.978+0	2.770+0	
266.0	1.028+1	9.191+0	6.291+0	4.651+0	4.265+0	3.926+0	3.624+0	3.356+0	3.119+0	2.906+0	2.715+0	

CONTRIBUTION BY EACH SOURCE TO THE HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 2 POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0
270.0	3.169+0	3.986+0	4.971+0	8.450+0	1.077+1	1.497+1	1.902+1	2.347+1	2.803+1	3.232+1	3.592+1
268.0	3.254+0	4.102+0	7.036+0	9.164+0	1.353+1	1.786+1	2.296+1	2.860+1	3.442+1	3.989+1	4.444+1
266.0	3.296+0	4.163+0	7.439+0	1.146+1	1.567+1	2.096+1	2.727+1	3.432+1	4.164+1	4.853+1	5.422+1

<< X(Km)

CONTRIBUTION BY EACH SOURCE TO THE HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 3 POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	1.428+0	1.499+0	1.568+0	1.629+0	1.685+0	1.425+0	1.454+0	1.473+0	1.485+0	1.488+0	1.483+0	
268.0	1.530+0	1.606+0	1.714+0	1.782+0	1.562+0	1.605+0	1.638+0	1.661+0	1.674+0	1.678+0	1.672+0	
266.0	1.634+0	1.716+0	1.869+0	1.694+0	1.755+0	1.806+0	1.843+0	1.870+0	1.885+0	1.889+0	1.882+0	

SECOND HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: ALL POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	1.453+1	1.433+1	1.502+1	1.547+1	1.757+1	2.017+1	2.281+1	2.575+1	2.881+1	3.177+1	3.439+1	
268.0	1.456+1	1.446+1	1.520+1	1.648+1	1.887+1	2.148+1	2.457+1	2.800+1	3.306+1	3.785+1	4.176+1	
266.0	1.459+1	1.454+1	1.522+1	1.729+1	1.964+1	2.261+1	2.819+1	3.477+1	4.169+1	4.824+1	5.355+1	

TIME OF SECOND HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (JULIAN DAY, HOUR)
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: ALL POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	193,14	193,12	193,12	193,13	193,11	193,12	193,12	193,12	193,12	193,12	193,12	
268.0	193,14	193,12	193,13	193,11	193,12	193,12	193,12	193,12	193,10	193,10	193,10	
266.0	193,14	193,12	193,13	193,12	193,12	193,12	193,10	193,10	193,10	193,10	193,10	

CONTRIBUTION BY EACH SOURCE TO THE SECOND HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 1 POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	1.119+1	7.817+0	6.917+0	7.699+0	4.698+0	4.942+0	4.467+0	4.060+0	3.712+0	3.410+0	3.148+0	
268.0	1.113+1	7.475+0	8.392+0	4.934+0	5.262+0	4.737+0	4.290+0	3.909+0	2.918+0	2.708+0	2.524+0	
266.0	1.109+1	7.131+0	8.212+0	5.591+0	5.006+0	4.512+0	3.376+0	3.118+0	2.893+0	2.695+0	2.521+0	

CONTRIBUTION BY EACH SOURCE TO THE SECOND HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194,07)
SOURCE: SOURCE 2 POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	2.111+0	5.005+0	6.537+0	6.135+0	1.149+1	1.350+1	1.657+1	1.989+1	2.327+1	2.652+1	2.940+1	
268.0	2.132+0	5.347+0	5.128+0	1.004+1	1.177+1	1.485+1	1.834+1	2.212+1	2.902+1	3.401+1	3.811+1	
266.0	2.132+0	5.615+0	5.214+0	9.759+0	1.262+1	1.603+1	2.354+1	3.035+1	3.749+1	4.423+1	4.972+1	

CONTRIBUTION BY EACH SOURCE TO THE SECOND HIGHEST 3-HOURLY CONCENTRATION (MICROGRAMS/M**3
FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 3 POLLUTANT: SO2 GRID: DINOSAUR

Y(Km)	246.0	248.0	250.0	252.0	254.0	256.0	258.0	260.0	262.0	264.0	266.0	<< X(Km)
270.0	1.222+0	1.503+0	1.568+0	1.634+0	1.388+0	1.733+0	1.772+0	1.802+0	1.825+0	1.838+0	1.843+0	
268.0	1.293+0	1.642+0	1.681+0	1.508+0	1.845+0	1.898+0	1.941+0	1.975+0	1.126+0	1.129+0	1.124+0	
266.0	1.364+0	1.789+0	1.796+0	1.945+0	2.014+0	2.073+0	1.275+0	1.296+0	1.308+0	1.311+0	1.306+0	

FIRST AND SECOND HIGHEST 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
AT INDIVIDUAL RECEPTORS FOR PERIOD: 07117807 (192,07) TO: 07137807 (194.07)
SOURCE: ALL POLLUTANT: SO2

RECEPTOR NAME	X-COORDINATE (KILOMETERS)	Y-COORDINATE (KILOMETERS)	HIGH CONCENTRATION	HIGH TIME	SECOND HIGH CONCENTRATION	SECOND HIGH TIME
SO FLT TOP	385.0	195.0	4.347+0	(193,22)	4.268+0	(193,23)
MEEKER	343.0	223.0	8.991+0	(193,19)	8.740+0	(193,20)
DINOSAUR	227.0	262.0	6.297+1	(194.07)	6.232+1	(194.06)

CONTRIBUTION BY EACH SOURCE TO THE 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
AT INDIVIDUAL RECEPTORS FOR PERIOD :07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 1 POLLUTANT: SO2

RECEPTOR NAME	X-COORDINATE (KILOMETERS)	Y-COORDINATE (KILOMETERS)	HIGH CONCENTRATION	HIGH TIME	SECOND HIGH CONCENTRATION	SECOND HIGH TIME
SO FLT TOP	385.0	195.0	1.914-1	(193,22)	2.168-1	(193,23)
MEEKER	343.0	223.0	5.466-1	(193,19)	7.443-1	(193,20)
DINOSAUR	227.0	262.0	6.254+1	(194,07)	6.197+1	(194,06)

CONTRIBUTION BY EACH SOURCE TO THE 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
AT INDIVIDUAL RECEPTORS FOR PERIOD :07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 2 POLLUTANT: SO2

RECEPTOR NAME	X-COORDINATE (KILOMETERS)	Y-COORDINATE (KILOMETERS)	HIGH CONCENTRATION	HIGH TIME	SECOND HIGH CONCENTRATION	SECOND HIGH TIME
SO FLT TOP	385.0	195.0	1.580+0	(193,22)	1.659+0	(193,23)
MEEKER	343.0	223.0	5.678+0	(193,19)	5.505+0	(193,20)
DINOSAUR	227.0	262.0	1.781-1	(194,07)	1.404-1	(194,06)

CONTRIBUTION BY EACH SOURCE TO THE 3-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)
AT INDIVIDUAL RECEPTORS FOR PERIOD :07117807 (192,07) TO: 07137807 (194.07)
SOURCE: SOURCE 3 POLLUTANT: SO2

RECEPTOR NAME	X-COORDINATE (KILOMETERS)	Y-COORDINATE (KILOMETERS)	HIGH CONCENTRATION	HIGH TIME	SECOND HIGH CONCENTRATION	SECOND HIGH TIME
SO FLT TOP	385.0	195.0	2.576+0	(193,22)	2.392+0	(193,23)
MEEKER	343.0	223.0	2.765+0	(193,19)	2.490+0	(193,20)
DINOSAUR	227.0	262.0	2.469-1	(194,07)	2.154-1	(194.06)

TABLE M-1

SUBROUTINE INTERACTION IN MET

Main Programs and Subroutines	Calling Routines																					
	AAMET	AMPL	AMPLIT	BETA	BLPARS	COEFF	CUP	FROUDE	LAPSE	LAY	MIXSTB	SRFC	SUBAO	TEMP	UPPER	UTM	VALGAM	WADJ	WARM	WINDS	WINGAM
AAMET[1]																					
ALTIM																					
AMPL			X																		
AMPLIT	X							X				X									
BETA																					
BLPARS	X					X															
COEFF	X																				
CUP																					
DESCNT						X															
DEW						X															
FILLMB																					
FILLME															X						
FINDU															X						
FROUDE			X			X					X		X	X	X						
GRID					X																
HGTS						X					X	X									
HPROF															X						
JULIAN	X								X		X				X						
LAPSE												X									
LAT			X					X							X						
LAY						X						X									
LLSQF[2]		X													X	X					
MBARS															X		X				
MIXSTB	X																				
MONIN															X						
NEAR					X																
POLBAS		X				X															
RDM																					
READS			X									X								X	X
READU			X		X										X						
SHSLP													X								
SMALLH		X																			
SOLAR	X							X			X									X	X
SRFC	X				X	X					X		X	X	X		X			X	X
STAB																					
SUBAO					X	X					X										
TEMP																					
TETRI			X				X														
THETA			X		X		X				X				X						
TPROF				X												X				X	
UPPER	X					X	X	X	X	X	X		X		X		X				
USTAR					X																
UTM			X			X															
VALGAM						X						X									
VREAL															X						
WADJ																					
WARM							X													X	X
WINDS					X								X								
WINGAM		X									X										
WNDRSP		X					X														
WSTAR				X			X														

[1] Main Program, [2] IMSL System Subroutine

TABLE M-2

COMMON BLOCKS IN MET

Main Programs and Subroutines	Common Blocks														
	A01	AMPL1	AMPLFN	FROUD	GRID1	LAPSE1	MAIN2	OTHER1	REINTS	REINTU	SRFC1	TEMP1	TETRI1	UPPER1	UPSFC1
AAMET[1]					X		X						X		
ALTIM															
AMPL		X	X												
AMPLIT		X	X				X	X			X			X	
BETA															
BLPARS			X		X		X				X				
COEFF						X								X	
CUP															
DESCNT															
DEW															
FILLMB															
FILLME															
FINDU															
FROUDE		X		X			X								
GRID															
HGTS															
HPROF															
JULIAN															
LAPSE				X		X						X		X	
LAT															
LAY															
LLSQF[2]															
MBARS															
MIXSTB			X		X		X				X			X	X
MONIN															
NEAR															
POLBAS															
RDM															
READS									X						
READU										X					
SHSLP															
SMALLH															
SOLAR															
SRFC											X				
STAB															
SUBAO	X										X			X	X
TEMP						X					X	X		X	
TETRI													X		
THETA															
TPROF															
UPPER							X							X	
USTAR															
UTM															
VALGAM															
VREAL															
WADJ		X		X											
WARM															
WINDS			X												
WINGAM			X												
WNDRSP															
WSTAR															

[1] Main Program, [2] IMSL System Subroutine

TABLE M-3

PARAMETER DECLARATIONS IN MET

Main Programs and Subroutines	Parameters																		
	NAL	NGA	NNS	NOB	NOBMX	NOT	NPF	NPG	NPO	NPT	NS	NSG	NTO	NUA	NVA	LREC	NREC	LRET	LREW	NRET	NREW
AAMET[1]		X						X		X											
ALTIM																					
AMPL	X	X							X												
AMPLIT	X	X	X			X			X					X							
BETA				X																	
BLPARS		X	X					X	X					X							
COEFF			X	X								X		X	X						
CUP				X																	
DESCNT											X										
DEW																					
FILLMB																					
FILLME																					
FINDU				X																	
FROUDE	X	X					X		X												
GRID																					
HGTS				X																	
HPROF				X																	
JULIAN																					
LAPSE				X			X							X							
LAT																					
LAY		X		X									X								
LLSQF[2]																					
MBARS				X																	
MIXSTB		X	X	X				X	X					X							
MONIN																					
NEAR														X							
POLBAS									X												
RDM																					
READS			X																		
READU		X												X							
SHSLP																					
SMALLH																					
SOLAR																					
SRFC																X	X				
STAB																					
SUBAO			X	X										X							
TEMP			X	X										X							
TETRI										X											
THETA																					
TPROF				X																	
UPPER		X		X	X									X				X	X	X	X
USTAR																					
UTM																					
VALGAM		X		X																	
VREAL									X												
WADJ	X	X					X														
WARM				X																	
WINDS		X							X												
WINGAM		X							X												
WNDRSP																					
WSTAR																					

[1] Main Program, [2] IMSL System Subroutine

TABLE M-4

PARAMETER DEFINITIONS AND VALUES

Parameter	Value	Description
NAL	155	NUA + NNS + NOT + (NPF x NPF)
NGA	9	Maximum number of 'gamma' surfaces
NNS	15	Maximum number of surface stations
NOB	30	2 x NOBMX
NOBMX	15	Maximum number of observations in soundings
NOT	30	Maximum number of wind interpolation points
NPF	10	Maximum number of points in X-direction or Y-direction on Froude grid
NPG	10	Maximum number of points in X-direction or Y-direction on output data grid
NPO	21	Maximum number of polynomial terms
NPT	50	Maximum number of points in X-direction or Y-direction on terrain grid
NS	60	Maximum number of time steps in subroutine DESCNT
NSG	10	Maximum number of segments per valley
NTO	39	NGA + NOB
NUA	10	Maximum number of upper air stations
NVA	5	Maximum number of valleys

TABLE M-5

SIZING OF DIRECT ACCESS DISK FILES
IN SUBROUTINES UPPER AND SRFC

Parameter	Value	Definition
NREC	10,800	NNS x (24 hr/day) x NDAYS* (Maximum number of surface weather data records)
LREC	9	Length of records (words)
NRET	600	NUA x (2 soundings/day) x NDAYS (Maximum number of upper-air temperature soundings)
NREW	600	NUA x (2 soundings/day) x NDAYS (Maximum number of upper-air wind soundings)
LRET	65	4 x NOBMX + 5 (length of temperature sounding records in words)
LREW	21	2 x NGA + 3 (length of wind sounding records in words)

* NDAYS = 30 (maximum number of days)

TABLE N-1

SUBROUTINE INTERACTION IN POLUT

Main Programs and Subroutines	Calling Routines																
	AAPOL	CALC	CALCVA	CONS	DIVIDE	INIT	NEWLOC	SIGGS	SIGMAV	SIGMAW	SIGMAY	SIGMAZ	UTM	VALY	WINDS	ZINITV
AAPOL[1]																
BLPINT	X						X	X						X		
CALC	X															
CALCVA		X	X													
CLEAN	X													X		
CLEANV	X															
CONS		X	X													
DEFINE		X	X													
DIVCK		X	X													
DIVIDE		X	X													
HTLOC				X												
INIT	X															
JULIAN						X										
LAT						X										
LITLR				X									X			
NEWLOC	X															
NOCALC						X								X		
OUT						X										
POLBAS																
PRISE	X														X	
ROUINT									X	X						
SHSLP																
SIGGS	X														X	
SIGMAV														X		
SIGMAW											X					
SIGMAY												X				X
SIGMAZ								X								
SMALLH								X								
TETRI					X	X	X				X				X	
UPDATE	X	X	X								X				X	
UTM						X								X		
VALMA	X															
VALY	X															
VERTDF				X												
VREAL																
WEIGHT			X		X										X	
WINDS	X						X	X			X			X		
YINIT	X															
YINITV														X		
YVIRT																
YVIRTG											X					
ZINIT	X										X					
ZINITV														X		
ZVIRT												X				

[1] Main Program

TABLE N-2

COMMON BLOCKS IN POLUT

Main Programs and Subroutines	Common Blocks																	
	BLPAR1	CLOCK1	CONC1	HTADJ	MAIN2	NAME1	NOC1	PUFF1	QUAIN	QUAOUT	REC	RGRIDC	RUNSP	SOURC1	TER1	TPDT	VALLY1	WINDS1
AAPOL[1]	X	X	X		X			X			X		X	X		X	X	X
BLPINT	X	X																
CALC								X								X		
CALCVA					X			X								X		
CLEAN								X										
CLEANV								X										
CONS			X				X			X	X							
DEFINE					X			X	X							X		
DIVCK									X		X	X						
DIVIDE									X	X								
HTLOC																		
INIT	X			X	X	X					X	X	X	X	X		X	X
JULIAN																		
LAT																		
LITLR																		
NEWLOC				X														
NOCALC							X				X			X				
OUT	X				X	X					X		X	X	X		X	
POLBAS																		
PRISE																		
ROUINT															X			
SHSLP																		
SIGGS								X					X			X		
SIGMAV																		
SIGMAW																		
SIGMAY					X													
SIGMAZ																		
SMALLH																		
TETRI					X										X			
UPDATE								X								X		
UTM																		
VALMA[4]														X			X	
VALY					X			X					X	X		X	X	
VERTDF																		
VREAL																		
WEIGHT																		
WINDS		X			X													X
YINIT																		
YINITV																		
YVIRT																		
YVIRTG																		
ZINIT																		
ZINITV																		
ZVIRT																		

[1] Main Program

TABLE N-3

PARAMETER DECLARATIONS IN POLUT

Main Programs and Subroutines	Parameters								
	IP	NGA	NPG	NPO	NPT	NPU	NSC	NSG	NVA
AAPOL[1]		X	X	X		X	X	X	X
BLPINT			X						
CALC						X	X	X	
CALCVA	X					X	X	X	
CLEAN						X	X	X	
CLEANV						X	X	X	
CONS	X						X		
DEFINE						X	X	X	
DIVCK									
DIVIDE	X								
HTLOC									
INIT		X	X	X	X		X	X	X
JULIAN									
LAT									
LITLR									
NEWLOC									
NOCALC							X		
OUT		X	X	X	X		X	X	X
POLBAS				X					
PRISE									
ROUINT					X				
SHSLP									
SIGGS						X	X	X	X
SIGMAV									
SIGMAW									
SIGMAY									
SIGMAZ									
SMALLH									
TETRI					X				
UPDATE						X	X	X	
UTM									
VALMA							X	X	X
VALY						X	X	X	X
VERTDF									
VREAL				X					
WEIGHT	X								
WINDS		X		X					
YINIT									
YINITV									
YVIRT									
YVIRTG									
ZINIT									
ZINITV									
ZVIRT									

[1] Main Program

TABLE N-4

PARAMETER DEFINITIONS AND VALUES

Parameter	Value	Description
IP	50	Maximum number of puffs on one side of a quadrilateral
NGA	9	Maximum number of 'gamma' surfaces
NPG	10	Maximum number of points in X-direction and Y-direction in gridded meteorological input data
NPO	21	Maximum number of polynomial terms
NPT	50	Maximum number of points in X-direction and Y-direction on terrain grid
NPU	100	Maximum number of puffs within domain for each source
NSC	5	Maximum number of sources
NSG	12	Maximum number of valley segments plus 2
NVA	5	Maximum number of valleys

APPENDIX O
FORTRAN LISTING OF TERRAIN

```
001     C     PROGRAM: TERRAIN.AATER        (VERSION 1.0)
002     C     DATE: APRIL 1985
003     C
004     C     THIS PROGRAM IS USED TO COMPUTE SMOOTHED TERRAIN HEIGHTS, AND
005     C   AVERAGE TERRAIN ROUGHNESS IN X-DIRECTION AND Y-DIRECTION.  TERRAIN
006     C   STATISTICS FOR THE FROUDE MODIFICATION GRID ARE ALSO COMPUTED.
007     C   THESE ARE THE LOWER TERRAIN HEIGHTS (AVERAGE HEIGHTS - STD. DEV.),
008     C   THE OBSTACLE HEIGHTS (2 * STD. DEV.), AND THE DRAINAGE DIRECTION.
009     C   THE DRAINAGE DIRECTION IS THE DIRECTION FROM WHICH THE DRAINAGE
010     C   WIND IS BLOWING (SPECIFIED AS DEGREES CLOCKWISE FROM NORTH).
011     C   THE INPUT DATA IS THE 500 BY 450 BASE TERRAIN FILE WHERE THE GRID
012     C   SPACING IS 1 KILOMETER.
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C   THE INPUTS ARE:
015     C     THRU LOGICAL UNIT 11 (FILE 'TOPO'):
016     C       ITHS = ARRAY OF THE 500X450 INPUT TERRAIN HEIGHTS (FT MSL)
017     C     THRU LOGICAL UNIT 10 (FILE 'TERIN'):
018     C       IDRUN = RUN ID NUMBER
019     C       THE FOLLOWING INPUTS ARE FOR TERRAIN SMOOTHING:
020     C         IX0 = GRAMA X-COORDINATE OF ORIGIN OF DOMAIN (KM)
021     C         IY0 = GRAMA Y-COORDINATE OF ORIGIN OF DOMAIN (KM)
022     C         NXC = NUMBER OF GRID CELLS IN X-DIR (<=50)
023     C         NYC = NUMBER OF GRID CELLS IN Y-DIR (<=50)
024     C         ICS = SIZE OF SQUARE GRID CELLS (KM)
025     C         IAI = SMOOTHING INTERVAL (KM)
026     C         LEN = LENGTH OF LINE SEGMENT FOR COMPUTING TERRAIN
027     C               ROUGHNESS (FOR MACCREADY'S = 14 KM)
028     C       THE FOLLOWING INPUTS ARE FOR FROUDE MODIFICATION GRID:
029     C         IXOF = GRAMA X-COORDINATE OF ORIGIN OF DOMAIN (KM)
030     C         IYOF = GRAMA Y-COORDINATE OF ORIGIN OF DOMAIN (KM)
031     C         NXCF = NUMBER OF GRID CELLS IN X-DIR (<=10)
032     C         NYCF = NUMBER OF GRID CELLS IN Y-DIR (<=10)
033     C         ICSF = SIZE OF SQUARE GRID CELLS (KM)
034     C         IAIF = SMOOTHING INTERVAL (KM)
035     C       CERTAIN CONSTRAINTS ARE:
036     C         1) ICS AND ICSF MUST BE GREATER THAN 1 (KM).
037     C         2) IAI MUST BE GREATER THAN OR EQUAL TO ICS.
038     C         3) IAIF MUST BE GREATER THAN OR EQUAL TO ICSF.
039     C         4) IF ICS IS AN EVEN NUMBER THEN IAI MUST BE EVEN,
040     C            IF ICS IS AN ODD NUMBER THEN IAI MUST BE ODD.
041     C         5) 4) ALSO HOLDS FOR ICSF AND IAIF.
042     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
043     C   EXAMPLE OF RUN SPECIFICATION FILE 'TERIN'
044     C       ' 04158512 '
045     C       0.0.50,45,10,20,14
046     C       0,0,10,9,50,50
047     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
048     C   THE OUTPUTS ARE:
049     C     THRU LOGICAL UNIT 19:
050     C       TAV = SMOOTHED TERRAIN HEIGHTS (M MSL)
051     C       TRX = AVERAGE ROUGHNESS IN X-DIRECTION (M)
052     C       TRY = AVERAGE ROUGHNESS IN Y-DIRECTION (M)
053     C     THRU LOGICAL UNIT 15:
054     C       TAVL = LOWER AVERAGED TERRAIN HEIGHTS [AVE - STD DEV] (M MSL)
055     C       OH = OBSTACLE HEIGHT [2*(STD. DEV. OF TERRAIN HEIGHTS)] (M)
056     C       THET = AVERAGE DRAINAGE DIRECTION (CLKWIS FROM N.)
057     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
058     C   FILES READ FROM:
059     C       LOGICAL UNIT 10 = SEQUENTIAL FORMATTED FILE OF RUN SPECS.
060     C       LOGICAL UNIT 11 = SEQUENTIAL UNFORMATTED FILE OF BASE TERRAIN.
061     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
062     C   FILES WRITTEN TO:
```

```
063     C       LOGICAL UNIT 6 = SUMMARY TABLE OF RUN RESULTS
064     C       LOGICAL UNIT 15 = SEQUENTIAL UNFORMATTED FILE OF FROUDE GRID
065     C                         OUTPUT
066     C       LOGICAL UNIT 19 = SEQUENTIAL UNFORMATTED FILE OF SMOOTHED
067     C                         TERRAIN DATA AND TERRAIN ROUGHNESS.
068     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
069     C  CALLED SUBROUTINES:
070     C       ROUGH
071     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
072     C  COMMON BLOCKS:
073     C       COM1 - SUBROUTINE ROUGH
074     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
075     C  PARAMETER DEFINITION:
076     C       NPT = MAX NUMBER OF OUTPUT GRID POINTS IS EITHER X OR Y DIR.
077     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
078     C
079           PARAMETER
080          +     NPT=50
081     C
082           CHARACTER*10
083          +     IDRUN
084     C
085           COMMON /COM1/
086          +     ITHS(500,450)
087     C
088           DIMENSION
089          +      ARR1(NPT),     ARR2(NPT),      ARR3(NPT)
090     C
091           RTOD=180./3.14159
092           LU=6
093     C
094     C               ***READ INPUT PARAMETERS.
095           READ (10,*) IDRUN
096           READ (10,*) IX0,IY0,NXC,NYC,ICS,IAI,LEN
097           WRITE (19) IX0,IY0,NXC,NYC,ICS.LEN
098           READ (10,*) IX0F,IY0F,NXCF,NYCF,ICSF,IAIF
099           WRITE (15) IX0F,IY0F,NXCF,NYCF,ICSF
100     C
101     C               ***READ IN TERRAIN DATA.
102           DO 1 J=1,450
103           READ (11) (ITHS(I,J),I=1,500)
104         1 CONTINUE
105     C
106     C               ***COMPUTE LOWER TERRAIN HEIGHT (AAR1), OBSTACLE HEIGHT
107     C               ***(ARR2), AND DRAINAGE DIRECTION (ARR3) FOR FROUDE
108     C               ***ADJUST GRID
109           IEXT=(IAIF-ICSF)/2
110           DO 4 M=1,NYCF
111             IBY=(IY0F+1)+ICSF*(M-1)-IEXT
112             IEY=IY0F+ICSF*M+IEXT
113             IF(IBY.LT.1) IBY=1
114             IF(IEY.GT.450) IEY=450
115             DO 3 L=1,NXCF
116               IBX=(IX0F+1)+ICSF*(L-1)-IEXT
117               IEX=IX0F+ICSF*L+IEXT
118               IF(IBX.LT.1) IBX=1
119               IF(IEX.GT.500) IEX=500
120               SUM=0.
121               SUMS=0.
122               ICOUNT=0
123     C             **COMPUTE TERRAIN AND OBJECT HEIGHT
124               DO 2 J=IBY,IEY
```

```
125               DO 2 I=IBX,IEX
126                 SUM=REAL(ITHS(I,J))+SUM
127                 SUMS=(REAL(ITHS(I,J)))**2+SUMS
128                 ICOUNT=ICOUNT+1
129         2     CONTINUE
130               COUNT=REAL(ICOUNT)
131               SDT=SQRT((SUMS-SUM**2/COUNT)/(COUNT-1))*.3048
132               ARR1(L)=(SUM/COUNT)*.3048-SDT
133               ARR2(L)=2.*SDT
134     C           **COMPUTE AVE SLOPE IN X-DIR
135               SUM=0.
136               ICOUNT=0
137               DEN=REAL(IEX-IBX)
138               DO 7 J=IBY,IEY
139                 SUM=REAL(ITHS(IEX,J)-ITHS(IBX,J))/DEN+SUM
140                 ICOUNT=ICOUNT+1
141         7     CONTINUE
142               B=SUM/REAL(ICOUNT)
143     C           **COMPUTE AVE SLOPE IN Y-DIR
144               SUM=0.
145               ICOUNT=0
146               DEN=REAL(IEY-IBY)
147               DO 8 I=IBX,IEX
148                 SUM=REAL(ITHS(I,IEY)-ITHS(I,IBY))/DEN+SUM
149                 ICOUNT=ICOUNT+1
150         8     CONTINUE
151               A=SUM/REAL(ICOUNT)
152     C             **COMPUTE DRAINAGE DIRECTION.
153               IF (B.EQ.0.) GO TO 11
154               ARR3(L)=ATAN(A/B)*RTOD
155               IF (B.LT.0.) ARR3(L)=ARR3(L)+180.
156               IF (B.LT.0.) GO TO 12
157               IF (A.LE.0.) ARR3(L)=ARR3(L)+360.
158               GO TO 12
159        11     IF (A.EQ.0.) ARR3(L)=999.
160               IF (A.LT.0.) ARR3(L)=270.
161               IF (A.GT.0.) ARR3(L)=90.
162        12     CONTINUE
163     C             **CHANGE DIRECTION CONVENTION FROM COUNTER-CLOCKWISE FROM
164     C             **X-AXIS(EAST-AXIS), TO CLOCKWISE FROM Y-AXIS(NORTH-AXIS)
165               IF (ARR3(L).GE.0..AND.ARR3(L).LE.90.) ARR3(L)=90.-ARR3(L)
166               IF (ARR3(L).GT.90..AND.ARR3(L).LE.360.) ARR3(L)=450.-ARR3(L)
167          3    CONTINUE
168               WRITE (15) (ARR1(L),L=1,NXCF)
169               WRITE (15) (ARR2(L),L=1,NXCF)
170               WRITE (15) (ARR3(L),L=1,NXCF)
171          4 CONTINUE
172     C
173     C**********************WRITE OUT SUMMARY TABLE*****************************
174           SLX=REAL(NXCF*ICSF)
175           SLY=REAL(NYCF*ICSF)
176           WRITE (LU,16)
177           WRITE (LU,17) IDRUN
178     C               ***WRITE OUT DOMAIN SPECIFICATIONS.
179           WRITE (LU,18)
180           WRITE (LU,19) SLX,SLY
181           XO=REAL(IXOF)
182           YO=REAL(IYOF)
183           XE=XO+SLX
184           YE=YO+SLY
185           WRITE (LU,20) XO,XE,YO,YE
186     C               ***WRITE OUT TERRAIN SPECS.
```

```
187           WRITE (LU,21)
188           WRITE (LU,33)
189           WRITE (LU,34) ICSF,ICSF,NXCF,NYCF,IAIF
190     C*************************************************************************
191     C
192     C                ***COMPUTE SMOOTHED TERRAIN HEIGHT (ARRAY ARR1)
193           IEXT=(IAI-ICS)/2
194           DO 24 M=1,NYC
195             IBY=(IY0+1)+ICS*(M-1)-IEXT
196             IEY=IY0+ICS*M+IEXT
197             IF(IBY.LT.1) IBY=1
198             IF(IEY.GT.450) IEY=450
199             DO 23 L=1,NXC
200               IBX=(IX0+1)+ICS*(L-1)-IEXT
201               IEX=IX0+ICS*L+IEXT
202               IF(IBX.LT.1) IBX=1
203               IF(IEX.GT.500) IEX=500
204               SUM=0.
205               ICOUNT=0
206     C            **SMOOTH TERRAIN
207               DO 22 J=IBY,IEY
208               DO 22 I=IBX,IEX
209                 SUM=REAL(ITHS(I,J))+SUM
210                 ICOUNT=ICOUNT+1
211          22     CONTINUE
212                 COUNT=REAL(ICOUNT)
213                 ARR1(L)=(SUM/COUNT)*.3048
214          23   CONTINUE
215               WRITE (19) (ARR1(L),L=1,NXC)
216          24 CONTINUE
217     C
218     C                ***COMPUTE TERRAIN ROUGHNESS IN X-DIR (ARR1) AND
219     C                ***Y-DIR (ARR2).
220           NPB=(LEN-1)/2
221           NPF=(LEN-1)-NPB
222           DO 40 M=1,NYC
223             IBY=(IY0+1)+ICS*(M-1)-IEXT
224             IEY=IY0+ICS*M+IEXT
225             IF(IBY.LT.1) IBY=1
226             IF(IEY.GT.450) IEY=450
227             DO 30 L=1,NXC
228               IBX=(IX0+1)+ICS*(L-1)-IEXT
229               IEX=IX0+ICS*L+IEXT
230               IF(IBX.LT.1) IBX=1
231               IF(IEX.GT.500) IEX=500
232     C              ***FIND TERRAIN ROUGHNESS IN X-DIR
233               IDIR=1
234               CALL ROUGH(IDIR,IBY,IEY,IBX,IEX,NPB,NPF,LEN,ROU)
235               ARR1(L)=ROU
236     C              ***FIND TERRAIN ROUGHNESS IN Y-DIR
237               IDIR=2
238               CALL ROUGH(IDIR,IBX,IEX,IBY,IEY,NPB,NPF,LEN,ROU)
239               ARR2(L)=ROU
240          30   CONTINUE
241               WRITE(19) (ARR1(L),L=1,NXC)
242               WRITE(19) (ARR2(L),L=1,NXC)
243          40 CONTINUE
244     C
245     C**********************WRITE OUT SUMMARY TABLE****************************
246           SLX=REAL(NXC*ICS)
247           SLY=REAL(NYC*ICS)
248           WRITE (LU,16)
```

```
      WRITE (LU,47) IDRUN
C               ***WRITE OUT DOMAIN SPECIFICATIONS.
      WRITE (LU,18)
      WRITE (LU,19) SLX,SLY
      X0=REAL(IX0)
      Y0=REAL(IY0)
      XE=X0+SLX
      YE=Y0+SLY
      WRITE (LU,20) X0,XE,Y0,YE
C               ***WRITE OUT TERRAIN SPECS.
      WRITE (LU,21)
      WRITE (LU,33)
      WRITE (LU,34) ICS,ICS,NXC,NYC,IAI
      WRITE (LU,35)
C*******************************************************************************
C
      STOP
   16 FORMAT (1H1,///,6X,'****SUMMARY RESULTS OF TERRAIN - TERRAIN',' PR
     +OCESSING PROGRAM****')
   17 FORMAT (1H0,5X,'RUN IDENTIFICATION: ',A10,'  (FROUDE GRID DATA)')
   47 FORMAT (1H0,5X,'RUN IDENTIFICATION: ',A10,'  (SMOOTH TERRAIN GRID'
     +,' DATA)')
   18 FORMAT (1H0,/,4X,'**DOMAIN**')
   19 FORMAT (1H0,5X,'REGION SIZE: ',F7.1,'  BY  ',F7.1,'  KILOMETERS')
   20 FORMAT (6X,'GRAMA COORDINATES (KM):   X= ',F7.1,'  TO  ',F7.1,/,30
     +X,'  Y= ',F7.1,'  TO  ',F7.1)
   21 FORMAT (1H0,/,4X,'**OUTPUT TERRAIN DATA**')
   33 FORMAT (6X,'AVERAGE TERRAIN STATISTICS WERE',' COMPUTED FROM THE B
     +ASE TERRAIN',/,6X,'FILE USING THE',' FOLLOWING GRID CELL SPECIFICA
     +TIONS:')
   34 FORMAT (11X,'CELL SIZE: ',I4,' BY ',I4,' KILOMETERS',/,11X,'NUMBER
     + OF CELLS IN X-DIR: ',I4,/,11X,'NUMBER OF CELLS IN Y-DIR: ',I4,
     +/,11X,'SMOOTHING INTERVAL: ',I4,' KILOMETERS')
   35 FORMAT (1H1)
      END
```

```
001     C  SUBROUTINE TERRAIN.ROUGH
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE AVERAGE TERRAIN ROUGHNESS
005     C  IN EITHER THE X-DIRECTION OR Y-DIRECTION FOR A GRID CELL.  THE
006     C  ROUGHNESS MEASURE IS THAT OF MACCREADY'S FOR HIS ANALYSIS OF THE
007     C  LO-LOCAT DATA.  IT IS BASED ON THE RMS ABOUT THE AVERAGE HEIGHT OF
008     C  A 14 KM SEGMENT.  A TERRAIN ROUGHNESS IS COMPUTED IN EITHER THE
009     C  X OR Y DIRECTION FOR EACH 1 KM POINT IN A GRID CELL.  THE TERRAIN
010     C  ROUGHNESS FOR ALL POINTS IN A GRID CELL ARE AVERAGED TOGETHER.
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C    THE INPUTS ARE:
013     C      DIR = 1 (X-DIR) OR 2 (Y-DIR)
014     C      IB1 = BEGINNING GRID POINT IN ONE DIRECTION
015     C      IE1 = ENDING GRID POINT IN ONE DIRECTION
016     C      IB2 = BEGINNING GRID POINT IN OTHER DIRECTION
017     C      IE2 = ENDING GRID POINT IN OTHER DIRECTION
018     C      NPB = # OF POINTS TO SUBTRACT TO GET START OF SEGMENT
019     C      NPF = # OF POINTS TO ADD TO GET END OF SEGMENT
020     C      LEN = SEGMENT LENGTH (M)
021     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
022     C    THE OUTPUT IS:
023     C      ROU = AVE TERRAIN ROUGHNESS FOR GRID CELL IN EITHER X OR Y (M)
024     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
025     C
026           SUBROUTINE ROUGH(IDIR,IB1,IE1,IB2,IE2,NPB,NPF,LEN,ROU)
027     C
028           COMMON /COM1/ ITHS(500,450)
029     C
030     C               ***DETERMINE ADJUSTMENTS TO INDICIES TO COMPUTE ROUGHNESS
031     C               ***AT 25 (5X5) GRID POINTS WITHIN A GRID CELL RATHER THAN
032     C               ***AT EVERY POINT
033           NP1=IE1-IB1+1
034           NPC1=NP1/5
035           NST1=((NP1-4*NPC1+1)/2)-1
036           NP2=IE2-IB2+1
037           NPC2=NP2/5
038           NST2=((NP2-4*NPC2+1)/2)-1
039     C
040           COUNT=0.
041           SIGTOT=0.
042           DO 20 J=IB1+NST1,IE1,NPC1
043           DO 20 I=IB2+NST2,IE2,NPC2
044           ISS=I-NPB
045           IES=I+NPF
046           COUNT=COUNT+1
047           ISUM=0
048           ISSUM=0
049     C          ***COMPUTE AVERAGE HEIGHT OF A SEGMENT
050           DO 10 K=ISS,IES
051           KK=K
052           IF(K.LE.0) KK=ABS(K-1)
053           IF(IDIR.EQ.1.AND.K.GT.500) KK=1001-K
054           IF(IDIR.EQ.2.AND.K.GT.450) KK=901-K
055           IF(IDIR.EQ.1) ISUM=ISUM+ITHS(KK,J)
056           IF(IDIR.EQ.2) ISUM=ISUM+ITHS(J,KK)
057        10 CONTINUE
058           IHAV=ISUM/LEN
059     C          ***COMPUTE RMS OF A SEGMENT
060           DO 5 K=ISS,IES
061           KK=K
062           IF(K.LE.0) KK=ABS(K-1)
```

```
063           IF(IDIR.EQ.1.AND.K.GT.500) KK=1001-K
064           IF(IDIR.EQ.2.AND.K.GT.450) KK=901-K
065           IF(IDIR.EQ.1) ISSUM=ISSUM +(ITHS(KK,J)-IHAV)**2
066           IF(IDIR.EQ.2) ISSUM=ISSUM +(ITHS(J,KK)-IHAV)**2
067         5 CONTINUE
068           SIG14=SQRT(REAL(ISSUM)/REAL(LEN))
069           SIGTOT=SIGTOT+SIG14
070        20 CONTINUE
071           ROU=SIGTOT/COUNT*.3048*2.3
072           RETURN
073           END
```

APPENDIX P
FORTRAN LISTING OF MET

```
001     C  PROGRAM: MET.AAMET       (VERSION 1.0)
002     C  DATE: APRIL 1985
003     C
004     C    THIS PROGRAM PRODUCES ANALYTICAL WIND FIELDS. GRIDDED MIXING
005     C  HEIGHTS, STABILITIES, TEMPERATURE, PRESSURE, FRICTION VELOCITIES,
006     C  MONIN-OBUKOV LENGTHS, AND CONVECTIVE VELOCITIES FOR EACH HOUR.
007     C  HOURLY VALLEY COUPLING COEFFICIENTS ARE ALSO COMPUTED FOR EACH
008     C  VALLEY SEGMENT.  THE INPUTS ARE DISK FILES CONTAINING RUN
009     C  SPECIFICATION INFORMATION, UPPER AIR SOUNDINGS, SURFACE OBSERVATIONS,
010     C  GRIDDED TERRAIN HEIGHTS, VALLEY CHARACTERISTICS (IF ANY VALLEYS),
011     C  AND GRIDDED LAND-USE CATEGORIES (IF AVAILABLE).
012     C  IF FROUDE NUMBER MODIFICATIONS ARE MADE TO THE FLOW FIELD THEN
013     C  A DISK FILE WHICH CONTAINS GRIDDED LOWER TERRAIN HEIGHTS, OBSTACLE
014     C  HEIGHTS, AND DRAINAGE DIRECTIONS ON A 10 BY 10 GRID ARE REQUIRED.
015     C    LIMITATIONS:
016     C      1. INPUT DATA LOCATIONS CANNOT BE MORE THAN TWO TIME ZONES APART.
017     C      2. THE BEGINNING HOUR MUST BE AFTER SUNRISE.
018     C      3. THE ENDING HOUR MUST BE BEFORE SUNSET.
019     C      4. CANNOT CROSS-OVER END OF YEAR OR USE HOUR 1 OF DAY JAN 1.
020     C      5. ANY DATA INPUT IN 00-TO-23 CLOCK IS CHANGED TO 01-TO-24
021     C         CLOCK BY DEFINING HOUR 00 AS HOUR 24 OF PREVIOUS DAY.
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  THE INPUTS ARE:
024     C    THRU LOGICAL UNIT 14 (FILE METIN):
025     C      IDRUN = A USER SPECIFIED RUN IDENTIFICATION (ALPHANUMERIC)
026     C      IBEG = SIMULATION BEGIN TIME IN BASE TIME ZONE (E.G. 08108012)
027     C      IEND = SIMULATION END TIME IN BASE TIME ZONE
028     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
029     C                              8(PACIFIC).
030     C      USENAM = USER NAME (CHARACTER*10)
031     C      DETAIL = .TRUE. MEANS OUTPUTS WRITTEN TO LOGICAL UNIT LUO
032     C      NSTA = NUMBER OF UPPER AIR STATIONS
033     C      NSSTA = NUMBER OF SURFACE STATIONS
034     C      NVALE = NUMBER OF VALLEYS
035     C      NOSTA = NUMBER OF OTHER LOCATIONS WHERE INPUT WIND PROFILES
036     C              ARE CONSTRUCTED
037     C      IXO = X-ORIGIN OF DOMAIN (KM)
038     C      IYO = Y-ORIGIN OF DOMAIN (KM)
039     C      SLX = DOMAIN SIZE IN X-DIRECTION (KM)
040     C      SLY = DOMAIN SIZE IN Y-DIRECTION (KM)
041     C      ALAT = APPROXIMATE LATITUDE OF GRID CENTER (DEG)
042     C      URLONG = APPROXIMATE LONGITUDE OF GRID CENTER (DEG)
043     C      NTEMAX = MAXIMUM # OF TERMS IN POLYNOMIAL LEAST-SQUARES FITS
044     C               OF 'BIG U' AND 'BIG V'
045     C               NTERMS=NDEG*(NDEG+1)/2+NDEG+1  (NDEG= DEGREE OF POLY.)
046     C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
047     C            1 - 'SMALL H' EQUAL TO CONSTANT
048     C            2 - 'BIG H' EQUAL TO CONSTANT
049     C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
050     C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
051     C           (SMALL H = BIG H - TERRAIN HEIGHT)
052     C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
053     C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
054     C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
055     C           TO HH)
056     C      NGAMA = NUMBER OF GAMMA SURFACES
057     C      HTGAMA = ARRAY OF GAMMA SURFACE HEIGHTS (0.0 TO 1.0)
058     C      FROU = IF TRUE SAYS WIND FIELD WILL BE ADJUSTED FOR FROUDE
059     C             NUMBER CONSIDERATIONS.
060     C      THE FOLLOWING 5 PARAMETERS GIVE THE GRID SPECIFICATION FOR
061     C      COMPUTING THE GRIDDED OUTPUTS:
062     C      IXOS = X ORIGIN OF COMPUTATIONAL DOMAIN (KM)
```

```
063     C       IYOS = Y ORIGIN OF COMPUTATIONAL DOMAIN (KM)
064     C       NXCS = NUMBER OF CELLS IN X-DIRECTION (UP TO 50)
065     C       NYCS = NUMBER OF CELLS IN Y-DIRECTION (UP TO 50)
066     C       ICSS = CELL SIZE (KM)
067     C       RAD = MAXIMUM SEPARATION DISTANCE, BETWEEN A SURFACE STATION
068     C             OR 'OTHER' STATION AND AN UPPER AIR STATION, AT WHICH
069     C             FOR ANY ACTUAL SEPARATION DISTANCE GREATER THAN 'RAD',
070     C             A 1/R**2 INTERPOLATION OF THE SOUNDING TO THE SURFACE
071     C             STATION OR 'OTHER' STATION WILL NOT BE PERFORMED (KM).
072     C       ZNOT = SURFACE ROUGHNESS (M), IF ZNOT = 999 THEN SURFACE
073     C              ROUGHNESS DETERMINED FROM GRIDDED LAND-USE CATEGORIES.
074     C       DS = SEED FOR RANDOM NUMBER GENERATOR (1.D0 TO 2147483647.D0).
075     C     THRU LOGICAL UNIT 19:
076     C       IXOT = X-ORIGIN OF INPUT TERRAIN DATA (KM)
077     C       IYOT = Y-ORIGIN OF INPUT TERRAIN DATA (KM)
078     C       NXCT = NUMBER OF CELLS IN X-DIRECTION OF TERRAIN FILE
079     C       NYCT = NUMBER OF CELLS IN Y-DIRECTION OF TERRAIN FILE
080     C       ICST = CELL SIZE OF TERRAIN FILE (KM)
081     C       THS = 2-D ARRAY OF TERRAIN HEIGHTS (M MSL)
082     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
083     C  FILES READ FROM:
084     C       LOGICAL UNIT 14 = FORMATTED SEQUENTIAL DISK FILE OF RUN
085     C                         SPECIFICATION DATA
086     C       LOGICAL UNIT 19 = UNFORMATTED SEQUENTIAL DISK FILE OF TERRAIN
087     C                         HEIGHTS
088     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
089     C  FILES WRITTEN TO:
090     C       LOGICAL UNIT LUO = FORMATTED SEQUENTIAL DISK FILE CONTAINING RUN
091     C                          OUTPUT INFORMATION
092     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
093     C  CALLED SUBROUTINES:
094     C       AMPLIT, BLPARS, COEFF, JULIAN, MIXSTB, SOLAR, SRFC, UPPER,
095     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
096     C  COMMON BLOCKS:
097     C       MAIN1, MAIN2, GRID1, TETRI1
098     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
099     C  PARAMETERS DEFINITIONS:
100     C       NGA = NUMBER OF 'GAMMA' SURFACES
101     C       NPT = NUMBER OF GRID POINTS ACROSS DOMAIN IN TERRAIN FILE
102     C       NPG = NUMBER OF GRID POINTS ACROSS DOMAIN FOR GRIDDED OUTPUT
103     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
104     C
105           PARAMETER
106          +    NGA=9,          NPT=50,         NPG=10
107     C
108           LOGICAL
109          +    DETAIL,         FROU
110     C
111           DIMENSION
112          +    HTGAMA(NGA)
113     C
114           DOUBLE PRECISION
115          +    DS
116     C
117           CHARACTER*10
118          +    TZNAM(8),       USENAM,         IDRUN
119     C
120           COMMON /MAIN2/
121          +    HH,             IHH,            IX0,            IY0,
122          +    RXY,            SLX,            SLY
123     C
124           COMMON /GRID1/
```

```
125           +    IXOS,          IYOS,          NXCS,          NYCS,
126           +    ICSS,          A(NPG),        B(NPG),        XR(NPG),
127           +    YR(NPG)
128      C
129            COMMON /TETRI1/
130           +    ICST,          IXOT,          IYOT,          NXCT,
131           +    NYCT,          THS(NPT,NPT)
132      C
133      C              ***SPECIFY TIME ZONE NAMES
134            DATA TZNAM/4*' ','    EST   ','    CST   ','    MST   ',
135           +                  '    PST   '/
136      C
137      C              ***SPECIFY LOGICAL UNIT FOR PRINTOUT
138            LUO=6
139      C
140      C               ***READ IN RUN INITIALIZATION DATA.
141            READ (14,*) IDRUN,IBEG,IEND,IBTZ,USENAM,DETAIL
142            READ (14,*) NSTA,NSSTA,NVALE,NOSTA
143            READ (14,*) IXO,IYO,SLX,SLY
144            READ (14,*) ALAT,URLONG
145            READ (14,*) NTEMAX,IHH,HH,NGAMA,(HTGAMA(K),K=1,NGAMA),FROU
146            READ (14,*) IXOS,IYOS,NXCS,NYCS,ICSS
147            READ (14,*) RAD,ZNOT,DS
148      C
149            ITZO=IBTZ
150            SXY=SLX/SLY
151      C
152      C              ***READ IN TERRAIN HEIGHTS.
153            READ (19) IXOT,IYOT,NXCT,NYCT,ICST
154            SSLX=REAL(NXCT*ICST)
155            SSLY=REAL(NYCT*ICST)
156            IF(IXOT.NE.IXO.OR.IYOT.NE.IYO.OR.SSLX.NE.SLX.OR.SSLY.NE.SLY)THEN
157      -       WRITE (6,4)
158              STOP 'AMAIN'
159            ENDIF
160            DO 1 J=1,NYCT
161              READ (19) (THS(L,J),L=1,NXCT)
162          1 CONTINUE
163      C
164      C              ***DETERMINE STARTING AND ENDING JULIAN DAY AND HOUR.
165            CALL JULIAN (IBEG,ITZO,IBTZ,IHS,IJS)
166            CALL JULIAN (IEND,ITZO,IBTZ,IHE,IJE)
167      C              **SET STARTING HOUR TO 2 HOURS AFTER SUNRISE IF
168      C              **INPUT STARTING HOUR IS EARLIER
169            TSN=30.
170            CALL SOLAR(IJS,TSN,ALAT,URLONG,IBTZ,A1,ALPHA,TSR,TSS)
171            ISR=NINT(TSR)
172            ISS=ISR+NINT(TSS-TSR)
173            IF(IHS.LT.ISR+2) IHS=ISR+2
174            IF(IHS.GT.ISS-2) IHS=ISS-2
175      C              **SET ENDING HOUR TO 2 HOURS BEFORE SUNSET IF INPUT
176      C              **ENDING HOUR IS LATER
177            IF(IJE.NE.IJS) THEN
178              TSN=30.
179              CALL SOLAR(IJE,TSN,ALAT,URLONG,IBTZ,A1,ALPHA,TSR,TSS)
180              ISR=NINT(TSR)
181              ISS=ISR+NINT(TSS-TSR)
182            ENDIF
183            IF(IHE.GT.ISS-2) IHE=ISS-2
184            IF(IHE.LT.ISR+2) IHE=ISR+2
185      C
186      C              ***COMPUTE GRID X AND Y AND ALPHA AND BETA COORDINATES
```

```
C              ***FOR OUTPUT GRID.
      XOS=REAL(IXOS)
      YOS=REAL(IYOS)
      STX=1./REAL(NXCS)
      A(1)=STX/2.
      XR(1)=A(1)*SLX+XOS
      DO 2 I=2,NXCS
        A(I)=A(I-1)+STX
        XR(I)=A(I)*SLX+XOS
    2 CONTINUE
      STY=1./REAL(NYCS)
      B(1)=STY/2.
      YR(1)=B(1)*SLY+YOS
      DO 3 J=2,NYCS
        B(J)=B(J-1)+STY
        YR(J)=B(J)*SLY+YOS
    3 CONTINUE
C
C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
      WRITE(LUO,5)
      WRITE(LUO,6)
      WRITE(LUO,6)
      WRITE(LUO,7)
      WRITE(LUO,6)
      WRITE(LUO,8)
      WRITE(LUO,6)
      WRITE(LUO,9) IDRUN,USENAM
      WRITE(LUO,10) IBEG,IJS,IHS
      WRITE(LUO,11) IEND,IJE,IHE
      WRITE(LUO,6)
      WRITE(LUO,12)
      WRITE(LUO,13)
      WRITE(LUO,14) IDRUN,IBEG,IEND,IBTZ,USENAM
      WRITE(LUO,15) DETAIL,NSTA,NSSTA,NVALE,NOSTA
      WRITE(LUO,16) IXO,IYO,SLX,SLY
      WRITE(LUO,17) ALAT,URLONG
      WRITE(LUO,18) NTEMAX,IHH,HH,NGAMA,FROU
      WRITE(LUO,19) RAD,ZNOT,DS
      WRITE(LUO,20) IXOS,IYOS,NXCS,NYCS,ICSS
      WRITE(LUO,21) (HTGAMA(K),K=1,NGAMA)
      WRITE(LUO,22) IXOT,IYOT,NXCT,NYCT,ICST
C*******************************************************************************
C
C              ***CALL SUBROUTINE TO PROCESS UPPER AIR DATA.
      CALL UPPER (NSTA,NGAMA,HTGAMA,DS,IBTZ,TZNAM(IBTZ),DETAIL,LUO)
C
C              ***CALL SUBROUTINE TO PROCESS SURFACE WEATHER DATA.
      CALL SRFC (NSSTA,DS,IBTZ,TZNAM(IBTZ),LUO)
C
C              ***CALL SUBROUTINE TO COMPUTE THE ANALYTICAL WIND FIELD
C              ***FOR EACH HOUR.
      CALL AMPLIT (IJS,IJE,IHS,IHE,RAD,NGAMA,HTGAMA,NSTA,NSSTA,NOSTA,
     +             NTEMAX,FROU,IBTZ,TZNAM(IBTZ),DETAIL,LUO)
C
C              ***CALL SUBROUTINE TO COMPUTE GRIDDED MIXING HEIGHT
C              ***AND STABILITY FIELDS FOR EACH HOUR.
      CALL MIXSTB (IJS,IJE,IHS,IHE,NSTA,NSSTA,NGAMA,HTGAMA,RAD,IBTZ,
     +             TZNAM(IBTZ),DETAIL,LUO)
C
C              ***CALL SUBROUTINE TO COMPUTE GRIDDED TEMP, PRESS, USTAR,
C              ***WSTAR, AND MONIN-OBUKOV LENGTH FOR EACH HOUR.
      CALL BLPARS(IJS,IJE,IHS,IHE,NSTA,NSSTA,NGAMA,HTGAMA,RAD,IBTZ,
```

```
     +            TZNAM(IBTZ),DETAIL,ZNOT,LUO)
C
C            ***CALL SUBROUTINE TO COMPUTE CUP/DECUP COEFFICIENTS
C            ***FOR EACH VALLEY SEGMENT FOR EACH HOUR.
      IF(NVALE.GT.0) THEN
      CALL COEFF(IJS,IJE.IHS,IHE,RAD,NSTA,NSSTA,NVALE,IBTZ,TZNAM(IBTZ),
     +           DETAIL,LUO)
      ENDIF
C
      STOP 'AAMAIN'
    4 FORMAT (2X,'DOMAIN SPECS FROM TERRAIN FILE NOT SAME AS RUN.')
    5 FORMAT(1H1,///,5X,10('**********'))
    6 FORMAT(1H ,4X,'*',98X,'*')
    7 FORMAT(1H ,4X,'*',43X,'PROGRAM MET',44X,'*',/,5X,'*',31X,
     +'+++METEOROLOGICAL DATA PROCESSOR+++',32X,'*',/,5X,'*',4X,
     +'PROVIDES HOURLY GRIDDED FIELDS OF MIXING HEIGHTS, STABILITIES,',
     +' TEMPERATURES, PRESSURES,',7X,'*',/,5X,'*',4X,'FRICTION',
     +' VELOCITIES, MONIN-OBUKOV LENGTHS, AND CONVECTIVE VELOCITIES.',
     +'  HOURLY 3-D ',12X,'*',/,5X,'*',4X,'ANALYTICAL WIND FIELDS ARE',
     +' PRODUCED IN ADDITION TO HOURLY VALLEY COUPLING COEFFICIENTS.',
     +7X,'*')
    8 FORMAT(1H ,4X,'*',98('+'),'*')
    9 FORMAT(1H ,4X,'*',6X,'THE RESULTS OF RUN ',A10,' ARE GIVEN.',25X,
     +'USER IS ',A10,9X,'*')
   10 FORMAT(1H ,4X,'*',6X,'PROCESSING START TIME IS (MODAYRHR) ',I8.8,
     +6X,'[ JULIAN DAY IS',I4,2X,'HOUR IS',I3,' ]',9X,'*')
   11 FORMAT(1H ,4X,'*',6X,'PROCESSING  STOP TIME IS (MODAYRHR) ',I8.8,
     +6X,'[ JULIAN DAY IS',I4,2X,'HOUR IS',I3,' ]',9X,'*')
   12 FORMAT(1H ,4X,10('**********'))
   13 FORMAT(1H0,//,5X,'THE RUN SPECIFICATIONS ARE (FILE METIN):')
   14 FORMAT(1H ,/,6X,'IDRUN = ',A10,2X,'IBEG = ',I8.8,5X,'IEND = ',
     +I8.8,5X,'IBTZ = ',I2,11X,'USENAM = ',A10)
   15 FORMAT(1H ,/,6X,'DETAIL = ',L2,9X,'NSTA = ',I3,10X,'NSSTA = ',I3,
     +9X,'NVALE = ',I3,9X,'NOSTA = ',I3)
   16 FORMAT(1H ,/,6X,'IX0 = ',I3,11X,'IY0 = ',I3,11X,'SLX = ',F5.0,9X,
     +'SLY = ',F5.0)
   17 FORMAT(1H ,/,6X,'ALAT = ',F7.3,6X,'URLONG = ',F7.3)
   18 FORMAT(1H ,/,6X,'NTEMAX = ',I3,8X,'IHH = ',I2,12X,'HH = ',F6.0,
     +9X,'NGAMA = ',I2,10X,'FROU = ',L2)
   19 FORMAT(1H ,/,6X,'RAD = ',F5.0,9X,'ZNOT = 'F6.2,7X,'DS = ',F5.0)
   20 FORMAT(1H ,/,6X,'IXOS = ',I3,10X,'IYOS = '.I3,10X,'NXCS = ',I2,
     +11X,'NYCS = '.I2,11X,'ICSS = ',I3)
   21 FORMAT(1H ,/,6X,'HTGAMA = '.9F10.4)
   22 FORMAT(1H0,//,5X,'THE TERRAIN GRID SPECIFICATIONS ARE (FROM',
     +' TERRAIN FILE):',//,6X,'IXOT = '.I3,10X,'IYOT = '.I3,10X,
     +'NXCT = ',I2,11X,'NYCT = '.I2,11X,'ICST = ',I3)
      END
```

```
001     C  SUBROUTINE: MET.ALTIM
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE DETERMINES SURFACE PRESSURE IN MILLIBARS
005     C  GIVEN THE ALTIMETER SETTING IN INCHES MERCURY AND THE
006     C  HEIGHT OF THE OBSERVATION IN METERS ABOVE MEAN SEA LEVEL.
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C  THE INPUTS ARE:
009     C       ZSFC = HEIGHT (M. MSL)
010     C       ALSTG = ALTIMETER SETTING (IN. HG.)
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE OUTPUTS ARE:
013     C       PSFC = SURFACE PRESSURE (MB)
014     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
015     C  CONSTANT DEFINITIONS:
016     C       C = 33.86389  [CONVERTS IN. HG. TO MB.]
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  CALLING SUBROUTINES:
019     C       SRFC
020     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
021     C  CALLED SUBROUTINES:
022     C       (NONE)
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  COMMON BLOCKS:
025     C       (NONE)
026     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
027     C
028           SUBROUTINE ALTIM (ZSFC,ALSTG,PSFC)
029     C
030           C1=(4.30899E-05)*ZSFC
031           C2=5.2553026
032           C3=1./C2
033           PSFC=((ALSTG**C3-C1)**C2)*33.86389
034           RETURN
035           END
```

```
001     C  SUBROUTINE: MET.AMPL
002     C  DATA: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE AMPLITUDE FUNCTIONS
005     C  'FI' AND 'GI' REFERENCED IN GRAMA REPORT 3 (AUG,1981) PAGE 82.
006     C  THESE FUNCTIONS ARE DETERMINED FOR EACH GAMMA SURFACE FROM THE
007     C  INPUT WIND PROFILE DATA AND ARE USED TO COMPUTE VELOCITIES AT
008     C  OTHER POINTS IN THE COMPUTATIONAL DOMAIN.  THE AMPLITUDE FUNCTIONS
009     C  FOR EACH 'GAMMA' SURFACE ARE COMPUTED AS A LEAST-SQUARES FIT TO
010     C  THE INPUT WINDS.
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE INPUTS ARE:
013     C    THRU ARGUMENT:
014     C      NTERMS = NUMBER OF POLYNOMIAL TERMS TO FIT WINDS
015     C      NSTA = NUMBER OF STATIONS
016     C      NGAMA = NUMBER OF 'GAMMA' SURFACES TO COMPUTE WINDS ON
017     C      HTGAMA = ARRAY OF HEIGHTS OF 'GAMMA' SURFACES (TRANSF. COORD)
018     C      HA = HEIGHT OF ADIABATIC ATMOSPHERE (M)
019     C      ROA = DENSITY AT SEA-LEVEL (KG/M**3)
020     C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
021     C            1 - 'SMALL H' EQUAL TO CONSTANT
022     C            2 - 'BIG H' EQUAL TO CONSTANT
023     C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
024     C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
025     C            (SMALL H = BIG H - TERRAIN HEIGHT)
026     C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
027     C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
028     C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
029     C           TO HH)
030     C      RXY = RATIO OF DOMAIN X-DIMENSION TO Y-DIMENSION
031     C    THRU COMMON BLOCK 'AMPL1':
032     C      UP = ARRAY OF X-COMP OF WINDS AT EACH 'GAMMA' SURFACE FOR ALL
033     C           INPUT WIND STATIONS (M/S)
034     C      VP = SAME AS 'UP' EXCEPT Y-COMP OF WINDS
035     C      XG = ARRAY OF 'ALPHA' COORD OF INPUT WIND STATIONS
036     C      YG = ARRAY OF 'BETA' COORD OF INPUT WIND STATIONS
037     C      ZSTER = ARRAY OF INPUT STATION ELEVATIONS (M MSL)
038     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
039     C  THE OUTPUTS ARE:
040     C    THRU COMMON BLOCK 'AMPLFN':
041     C      FI = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR EACH 'GAMMA'
042     C           SURFACE FROM INPUT WINDS
043     C      GI = ARRAY OF AMPLITUDE FUNCTION 'GSUGI' FOR EACH 'GAMMA'
044     C           SURFACE FROM INPUT WINDS
045     C      GMF = ARRAY OF AMPLITUDE FUNCTION 'GSUBI-FSUBI' FOR EACH 'GAMMA'
046     C            SURFACE FROM INPUT WINDS
047     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
048     C  CALLING SUBROUTINES:
049     C      AMPLIT, FROUDE
050     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
051     C  CALLED SUBROUTINES:
052     C      POLBAS, SMALLH, LLSQF(IMSL SUBROUTINE)
053     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
054     C  COMMON BLOCKS:
055     C      AMPL1. AMPLFN
056     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
057     C  PARAMETERS DEFINITIONS:
058     C      NAL = NUA+NNS+NOT+NFR [TOTAL # STA TO COMPUTE AMPL FNCTS,
059     C            NUA = NUMBER OF UPPER AIR STATIONS,
060     C            NNS = NUMBER OF SURFACE STATIONS,
061     C            NOT = NUMBER OF 'OTHER' STATIONS,
062     C            NFR = # OF GRID POINTS FOR FROUDE NUMBER ADJUSTMENT]
```

```
C       NGA = NUMBER OF 'GAMMA' SURFACES
C       NPO = NUMBER OF POLYNOMIAL TERMS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE AMPL (NTERMS,NSTA,NGAMA,HTGAMA,HA.ROA,HH,IHH,RXY)
C
      PARAMETER
     +    NAL=155,        NGA=9,          NPO=21
C
      DIMENSION
     +    BI(NPO),        CI(NPO),        H(NPO),         HTGAMA(NGA),
     +    IP(NPO),        PBI(NPO),       UST(NAL),       VST(NAL),
     +    XU(NPO),        XV(NPO),        XXA(NAL,NPO),   XXB(NAL,NPO)
C
      COMMON /AMPL1/
     +    UP(NAL,NGA),    VP(NAL,NGA),    XG(NAL),        YG(NAL),
     +    ZSTER(NAL)
C
      COMMON /AMPLFN/
     +    FI(NPO,NGA),    GI(NPO,NGA),    GMF(NPO,NGA)
C
C         ***COMPUTE AMPLITUDE FUNCTIONS.
      DO 4 J=1,NGAMA
      DO 2 NS=1.NSTA
C
C               ***COMPUTE 'BIG U' AND 'BIG V'.
      CALL SMALLH (IHH,HH,ZSTER(NS),XG(NS),YG(NS),HZS)
      Z=ZSTER(NS)+HTGAMA(J)*HZS
      RO=ROA*(1.-Z/HA)**2.5
      ROZ=RO*HZS
      UST(NS)=ROZ*UP(NS,J)
      VST(NS)=RXY*ROZ*VP(NS,J)
C
C               ***COMPUTE POLYNOMIAL BASIS FUNCTIONS FOR EACH STATION.
      CALL POLBAS (XG(NS),YG(NS),NTERMS,BI,CI,PBI)
      DO 1 IC=1,NTERMS
      XXA(NS,IC)=BI(IC)
      XXB(NS,IC)=CI(IC)
    1 CONTINUE
C
    2 CONTINUE
C
C               ***CALL LINEAR LEAST SQUARES FIT SUBROUTINE (IMSL).
      TOL=-1.0
      M=NSTA
      IA=NAL
      NNA=NTERMS
      KBASIS=NTERMS
      CALL LLSQF (XXA.IA,M,NNA.UST,TOL,KBASIS,XU,H,IP,IER)
      TOL=-1.0
      KBASIS=NTERMS
      CALL LLSQF (XXB,IA,M,NNA.VST,TOL,KBASIS,XV,H,IP,IER)
C
      DO 3 I=1.NTERMS
      FI(I,J)=XU(I)
      GI(I,J)=XV(I)
    3 CONTINUE
C
    4 CONTINUE
C
C               ***EVALUATE INTEGRAL OF 'GI-FI' USED IN COMPUTING 'W'.
      DO 5 IC=1.NTERMS
```

```
125           DO 5 J=1,NGAMA
126           IF(J.EQ.1) THEN
127           GMF(IC,J)=(GI(IC,J)-FI(IC,J))*HTGAMA(J)
128           ELSE
129           GMF(IC,J)=GMF(IC,J-1)+(GI(IC,J)-FI(IC,J))*(HTGAMA(J)-HTGAMA(J-1))
130           ENDIF
131         5 CONTINUE
132     C
133           RETURN
134           END
```

```
001     C  SUBROUTINE: MET.AMPLIT
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE AMPLITUDE FUNCTIONS FOR
005     C  THE MASS-CONSISTENT WIND FIELD GIVEN THE INPUT UPPER AIR AND
006     C  SURFACE WEATHER DATA.  THE AMPLITUDE FUNCTIONS ARE COMPUTED
007     C  USING A LEAST-SQUARES FIT TO THE INPUT WINDS ON PREDEFINED
008     C  'SURFACES'.  CHANNELING AROUND MAJOR TERRAIN FEATURES CAN BE
009     C  ACCOUNTED FOR USING A FROUDE NUMBER MODIFICATION SCHEME.
010     C  THE OUTPUT IS AN UNFORMATTED DISK FILE WRITTEN TO UNIT 12 WHICH
011     C  CONTAINS THE AMPL FNCTNS FOR EACH SRFC FOR EACH HOURLY TIME STEP.
012     C  THE AMPLITUDE FUNCTIONS FOR EACH TIME STEP ARE WRITTEN OUT SO
013     C  THAT EACH RECORD CONTAINS THE AMPLITUDE FUNCTIONS FOR ONE SURFACE.
014     C  FOR EXAMPLE, IF A SIMULATION IS DONE FOR 4 SURFACES, THIS MEANS
015     C  THAT THE OUTPUT FOR EACH TIME STEP WILL CONTAIN 4 RECORDS.
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  THE INPUTS ARE:
018     C    THRU ARGUMENT:
019     C      IJS = STARTING JULIAN DATE
020     C      IJE = ENDING JULIAN DATE
021     C      IHS = STARTING HOUR-OF-DAY (1 TO 24)
022     C      IHE = ENDING HOUR-OF-DAY (1 TO 24)
023     C      RAD = MAXIMUM SEPARATION DISTANCE, BETWEEN A SURFACE OR 'OTHER'
024     C            STATION AND AN UPPER AIR STATION, AT WHICH FOR ANY ACTUAL
025     C            SEPARATION DISTANCE GREATER THAN 'RAD', A 1/R**2
026     C            INTERPOLATION OF THE SOUNDING TO THE SURFACE OR 'OTHER'
027     C            STATION WILL NOT BE PERFORMED (KM).
028     C      NGAMA = NUMBER OF 'GAMMA' SURFACES
029     C      HTGAMA = ARRAY OF HEIGHT OF GAMMA SURFACES (GAMMA COORD)
030     C      NSTA = NUMBER OF UPPER AIR STATIONS
031     C      NSSTA = NUMBER OF SURFACE STATIONS
032     C      NOSTA = NUMBER OF 'OTHER' STATIONS
033     C      NTEMAX = NUMBER OF POLYNOMIAL TERMS TO FIT WINDS
034     C      FROU = IF TRUE MEANS ADJUST WIND FIELD ACCORDING TO FROUDE #.
035     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
036     C                              8(PACIFIC)
037     C      TZNAM = BASE TIME ZONE NAME
038     C      DETAIL = LOGICAL VARIABLE IF .TRUE. THEN WRITE OUT INFORMATION
039     C               TO UNIT LUO
040     C      LUO = LOGICAL UNIT TO WHICH PRINTOUT IS DIRECTED
041     C    THRU LOGICAL UNIT 4:
042     C      IDO = ARRAY OF OTHER STATION ID NUMBERS
043     C      NAMEO = ARRAY OF NAMES OF OTHER STATIONS
044     C      XU = X-COORD OF OTHER STATIONS
045     C      YU = Y-COORD OF OTHER STATIONS
046     C      ICODE = CODE SPECIFYING COORDINATE SYSTEM
047     C            =  1 [LATITUDE-LONGITUDE]
048     C            =  2 [UTM ZONE 12]
049     C            =  3 [UTM ZONE 13]
050     C            =  4 [GRAMA]
051     C    THRU COMMON BLOCK 'MAIN2':
052     C      IXO = X-COORD OF DOMAIN ORIGIN IN GRAMA COORDS (KM)
053     C      IYO = Y-COORD OF DOMAIN ORIGIN IN GRAMA COORDS (KM)
054     C      SLX = DOMAIN WIDTH IN X-DIR (KM)
055     C      SLY = DOMAIN WIDTH IN Y-DIR (KM)
056     C    THRU COMMON BLOCK 'UPPER1':
057     C      COX = ARRAY OF X-COORD OF UPPER AIR STATIONS (KM)
058     C      COY = ARRAY OF Y-COORD OF UPPER AIR STATIONS (KM)
059     C      EL = ARRAY OF UPPER AIR STATIONS ELEVATIONS (M MSL)
060     C      IDU = ARRAY OF UPPER AIR STATIONS IDENTIFICATIONS
061     C      NSDW = ARRAY OF NUM OF WIND SNDGS FOR UPPER AIR STATIONS
062     C      IUSRTW = ARRAY OF STARTING RECORD NUMBER OF WIND SNDGS FOR ALL
```

```
063     C                UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 8
064     C     THRU COMMON BLOCK 'SRFC1':
065     C       ELS = ARRAY OF SURFACE STATIONS ELEVATION (M MSL)
066     C       IDS = ARRAY OF SURFACE STATIONS ID NUMBERS
067     C       NOBS = ARRAY OF NUM OF OBS FOR SURFACE STATIONS
068     C       SCOX = ARRAY OF X-COORD OF SURFACE STATIONS (KM)
069     C       SCOY = ARRAY OF Y-COORD OF SURFACE STATIONS (KM)
070     C       ISSRT = ARRAY OF STARTING RECORD NUMBER OF SRFC OBS FOR ALL
071     C                SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
072     C     THRU COMMON BLOCK 'AMPLFN':
073     C       FI = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR EACH 'GAMMA'
074     C            SURFACE FROM INPUT AND ADJUSTED SOUNDINGS
075     C       GI = ARRAY OF AMPLITUDE FUNCTION 'GSUGI' FOR EACH 'GAMMA'
076     C            SURFACE FROM INPUT AND ADJUSTED SOUNDINGS
077     C       GMF = ARRAY OF AMPLITUDE FUNCTION 'GSUBI-FSUBI' FOR EACH 'GAMMA'
078     C             SURFACE FROM INPUT AND ADJUSTED SOUNDINGS
079     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
080     C  THE OUTPUTS ARE:
081     C     THRU COMMON BLOCK 'AMPL1':
082     C       UP = ARRAY OF X-COMP OF WINDS AT EACH 'GAMMA' SURFACE FOR ALL
083     C            INITIAL INPUT WIND STATIONS (M/S)
084     C       VP = SAME AS 'UP' EXCEPT Y-COMP OF WINDS
085     C       XG = ARRAY OF 'ALPHA' COORD OF INPUT WIND STATIONS
086     C       YG = ARRAY OF 'BETA' COORD OF INPUT WIND STATIONS
087     C       ZSTER = ARRAY OF INPUT STATION ELEVATIONS (M MSL)
088     C     THRU LOGICAL UNIT 12:
089     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
090     C  FILES READ FROM:
091     C       LOGICAL UNIT 4 = SEQUENTIAL FORMATTED DISK FILE CONTAINING
092     C                        DATA ON THE 'OTHER' INPUT STATIONS.
093     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
094     C  FILES WRITTEN TO:
095     C       LOGICAL UNIT 12 = UNFORMATTED SEQUENTIAL DISK FILE CONTAINING
096     C                         AMPLITUDE FUNCTIONS FOR EACH 'GAMMA' SRFC
097     C       LOGICAL UNIT LUO = IF DETAIL IS .TRUE. THEN RUN INFO IS OUTPUT
098     C       LOGICAL UNIT  6 = ERROR MESSAGES
099     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
100     C  CALLING SUBROUTINES:
101     C       AAMAIN
102     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
103     C  CALLED SUBROUTINES:
104     C       AMPL, FROUDE, LAT, READS, READU, THETA, UTM, WINGAM, WNDRSP,
105     C       TETRI
106     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
107     C  COMMON BLOCKS:
108     C       MAIN2, UPPER1, SRFC1, AMPL1, AMPLFN, OTHER1
109     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
110     C  PARAMETERS DEFINITIONS:
111     C       NAL = NUA+NNS+NOT+(NPF*NPF) [TOTAL # STA TO COMPUTE AMPL FNCTS,
112     C             NPF*NPF = # OF GRID POINTS FOR FROUDE NUMBER ADJUSTMENT]
113     C       NGA = NUMBER OF 'GAMMA' SURFACES
114     C       NNS = NUMBER OF SURFACE STATIONS
115     C       NOT = NUMBER OF 'OTHER' STATIONS
116     C       NUA = NUMBER OF UPPER AIR STATIONS
117     C       NPO = NUMBER OF POLYNOMIAL TERMS
118     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
119     C
120           SUBROUTINE AMPLIT (IJS,IJE,IHS,IHE,RAD,NGAMA,HTGAMA,NSTA.NSSTA.
121          +                   NOSTA,NTEMAX,FROU,IBTZ,TZNAM.DETAIL,LUO)
122     C
123           PARAMETER
124          +    NAL=155,       NGA=9,         NNS=15,        NOT=30,
```

```
125           +    NPO=21,          NUA=10
126     C
127            LOGICAL
128           +    DETAIL,          PASS,            FRO1ST,          FROU,
129           +    RDU1ST,          RDS1ST
130     C
131            DIMENSION
132           +    COX1(NUA),       COY1(NUA),       HTGAMA(NGA),
133           +    IDU1(NUA),       NOBSS(NNS),      NSND(NUA),       OWIX(NOT,NGA),
134           +    OWIY(NOT,NGA),   OWRA(NOT),       WIX(NNS,NGA),    WIY(NNS,NGA),
135           +    WRA(NNS),        WX(NGA),         WXU(NUA,NGA),    WY(NGA),
136           +    WYU(NUA,NGA),    WDI(NGA),        WDO(NGA),        WSI(NGA),
137           +    WSO(NGA),        WZ(NGA)
138     C
139            CHARACTER*10
140           +    NAME,            NAMEO,           NAMES,           TZNAM
141     C
142            COMMON /MAIN2/
143           +    HH,              IHH,             IXO,             IYO,
144           +    RXY,             SLX,             SLY
145     C
146            COMMON /UPPER1/
147           +    COX(NUA),        COY(NUA),        EL(NUA),         IDU(NUA),
148           +    NAME(NUA),       NSDT(NUA),       NSDW(NUA),       XLAT(NUA),
149           +    YLONG(NUA),      IUSRTW(NUA),     IUSRTT(NUA),     IUENDW(NUA),
150           +    IUENDT(NUA),     ITZU(NUA)
151     C
152            COMMON /SRFC1/
153           +    ELS(NNS),        IDS(NNS),        NAMES(NNS),      NOBS(NNS),
154           +    SCOX(NNS),       SCOY(NNS),       SLAT(NNS),       SLONG(NNS),
155           +    ISSRT(NNS),      ISEND(NNS)
156     C
157            COMMON /AMPL1/
158           +    UP(NAL,NGA),     VP(NAL,NGA),     XG(NAL),         YG(NAL),
159           +    ZSTER(NAL)
160     C
161            COMMON /AMPLFN/
162           +    FI(NPO,NGA),     GI(NPO,NGA),     GMF(NPO,NGA)
163     C
164            COMMON /OTHER1/
165           +    IDO(NOT),        NAMEO(NOT),      OCOX(NOT),       OCOY(NOT)
166     C
167            FRO1ST=.TRUE.
168            RDU1ST=.TRUE.
169            RDS1ST=.TRUE.
170     C
171     C             ***WRITE FILE ID NUMBER TO UNIT 12.
172            WRITE (12) IXO,IYO,SLX,SLY
173            WRITE (12) NGAMA,(HTGAMA(KK),KK=1,NGAMA),IHH,HH
174     C
175     C             ***READ IN OTHER STATION INFORMATION (MAX 20 STATIONS).
176            IF(NOSTA.EQ.0) GO TO 6
177            DO 6 J=1,NOSTA
178            READ (4,35) IDO(J),NAMEO(J),XU,YU,ICODE
179            GO TO (4,3,3,5), ICODE
180          3 IZONE=ICODE
181            CALL UTM (IZONE,XU,YU,OCOX(J),OCOY(J))
182            GO TO 6
183          4 CALL LAT (XU,YU,OCOX(J),OCOY(J))
184            GO TO 6
185          5 OCOX(J)=XU
186            OCOY(J)=YU
```

```
      6 CONTINUE
C
C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
       WRITE(LUO,40) TZNAM
       WRITE(LUO,41) (HTGAMA(NG),NG=1,NGAMA)
       IF(IHH.EQ.1) WRITE(LUO,48) HH
       IF(IHH.EQ.2) WRITE(LUO,50) HH
       WRITE(LUO,39) NTEMAX
       WRITE(LUO,42)
       DO 7 J=1,NOSTA
         WRITE(LUO,43) NAMEO(J),IDO(J),OCOX(J),OCOY(J)
     7 CONTINUE
C*******************************************************************************
C
C                ***LOOP ON JULIAN DAY AND HOUR OF DAY.
       DO 500 JUL=IJS,IJE
       ICLKP=(JUL-1)*24
       ISA=1
       ISP=24
       IF(JUL.EQ.IJS) ISA=IHS
       IF(JUL.EQ.IJE) ISP=IHE
C
C                ***LOOP ON HOUR OF DAY
       DO 500 IHR=ISA,ISP
       ICLK=ICLKP+IHR
C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
       IF(DETAIL) THEN
         WRITE(LUO,44) JUL,IHR
       ENDIF
C*******************************************************************************
C
C                ***INITIALIZE REQUIRED VARIABLES.
       DO 8 K=1,NSSTA
         WRA(K)=0.
         DO 8 KK=1,NGAMA
           WIX(K,KK)=0.
           WIY(K,KK)=0.
     8 CONTINUE
       IF(NOSTA.EQ.0) GO TO 9
       DO 9 K=1,NOSTA
         OWRA(K)=0.
           DO 9 KK=1,NGAMA
           OWIX(K,KK)=0.
           OWIY(K,KK)=0.
     9 CONTINUE
       NUSTA=0
C
C              ***LOOP ON UPPER AIR STATIONS COMPUTING 1/R**2 WEIGHTED
C              ***SOUNDINGS TO EACH SURFACE AND "OTHER" STATION.
       DO 19 JJ=1,NSTA
       CALL READU (ICLK,NSND(JJ),IUSRTW(JJ),NSDW(JJ),NGAMA,JJ,RDU1ST,
      +            PASS,WX,WY)
       IF (PASS) GO TO 19
       NUSTA=NUSTA+1
       COX1(NUSTA)=COX(JJ)
       COY1(NUSTA)=COY(JJ)
       IDU1(NUSTA)=IDU(JJ)
       DO 12 L=1,NGAMA
         WXU(NUSTA,L)=WX(L)
         WYU(NUSTA,L)=WY(L)
    12 CONTINUE
C
```

```
C              ***WEIGHT THE UPPER AIR SOUNDING TO EACH SURFACE STATION.
      DO 15 K=1,NSSTA
        RDIF=((COX(JJ)-SCOX(K))**2.+(COY(JJ)-SCOY(K))**2.)**.5
        IF (RDIF.GT.RAD) GO TO 15
        IF (RDIF.LE.0.1.AND.IDS(K).NE.IDU(JJ))  THEN
          WRITE (6,36)
          STOP 'AMPLIT'
        ENDIF
        IF (RDIF.LE.0.1) RDIF=1.*10.**(-3.)
        WRA(K)=1./RDIF**2.+WRA(K)
        DO 14 KK=1,NGAMA
          WIX(K,KK)=(1./RDIF**2.)*WXU(NUSTA,KK)+WIX(K,KK)
          WIY(K,KK)=(1./RDIF**2.)*WYU(NUSTA,KK)+WIY(K,KK)
   14   CONTINUE
   15 CONTINUE
C
C              ***WEIGHT THE UPPER AIR SOUNDING TO EACH 'OTHER' STATION.
      IF(NOSTA.EQ.0) GO TO 17
      DO 17 K=1,NOSTA
        RDIF=((COX(JJ)-OCOX(K))**2.+(COY(JJ)-OCOY(K))**2.)**.5
        IF (RDIF.GT.RAD) GO TO 17
        IF (RDIF.LE.0.1) RDIF=1.*10.**(-3.)
        OWRA(K)=1./RDIF**2.+OWRA(K)
        DO 16 KK=1,NGAMA
          OWIX(K,KK)=(1./RDIF**2.)*WXU(NUSTA,KK)+OWIX(K,KK)
          OWIY(K,KK)=(1./RDIF**2.)*WYU(NUSTA,KK)+OWIY(K,KK)
   16   CONTINUE
   17 CONTINUE
C
   19 CONTINUE
      RDU1ST=.FALSE.
C
C              ***COMPUTE 1/R**2 AVERAGE WINDS FOR EACH SURFACE STATION.
      DO 20 K=1,NSSTA
      DO 20 KK=1,NGAMA
        WIX(K,KK)=WIX(K,KK)/WRA(K)
        WIY(K,KK)=WIY(K,KK)/WRA(K)
   20 CONTINUE
C
C              ***INPUT NEW SURFACE WIND IF DATA GIVEN.
      ITHETC=0
      THTTAS=0.
      DO 24 K=1,NSSTA
        CALL READS (ICLK,NOBSS(K),ISSRT(K),NOBS(K),K,RDS1ST,PASS,WXS,
     +              WYS,TSURF,RHO,ICC)
        IF (PASS) GO TO 24
        WIX(K,1)=WXS
        WIY(K,1)=WYS
C              ***COMPUTE THETA IN DRAKE'S DENSITY RELATIONSHIP
        CALL THETA(RHO,ELS(K),THTTA)
        THTTAS=THTTAS+THTTA
        ITHETC=ITHETC+1
   24 CONTINUE
      RDS1ST=.FALSE.
C              ***COMPUTE AVERAGE THETA FROM SRFC. OBS. FOR TIME STEP,
C              ***AND COMPUTE CONSTANTS USED IN DENSITY CALCULATION
      IF (ITHETC.EQ.0) THEN
        THETTA=300.
      ELSE
        THETTA=THTTAS/FLOAT(ITHETC)
      ENDIF
      ROA=353.1/THETTA
```

```
311           HA=102.4*THETTA
312     C
313     C                ***COMPUTE 1/R**2 AVERAGE WINDS FOR EACH OTHER STATION.
314           IF(NOSTA.EQ.0) GO TO 25
315           DO 25 K=1,NOSTA
316             DO 25 KK=1,NGAMA
317               OWIX(K,KK)=OWIX(K,KK)/OWRA(K)
318               OWIY(K,KK)=OWIY(K,KK)/OWRA(K)
319        25 CONTINUE
320     C
321     C                ***COMPUTE AMPLITUDE FUNCTIONS FOR TIME STEP USING
322     C                ***SOUNDINGS FROM UPPER AIR STATIONS, SURFACE AND OTHER
323     C                ***STATIONS.
324           NS=0
325           XO=REAL(IXO)
326           YO=REAL(IYO)
327           DO 27 K=1,NUSTA
328           IF (COX1(K).LT.XO.OR.COX1(K).GT.XO+SLX.OR.COY1(K).LT.YO.OR.
329          +    COY1(K).GT.YO+SLY) GO TO 27
330     C                ***CHECK TO SEE IF HAVE SURFACE STATION WITH SAME ID
331     C                ***NUMBER.
332           DO 26 ICHK=1,NSSTA
333             IF (IDU1(K).EQ.IDS(ICHK)) GO TO 27
334        26 CONTINUE
335           NS=NS+1
336           XG(NS)=(COX1(K)-XO)/SLX
337           YG(NS)=(COY1(K)-YO)/SLY
338           CALL TETRI(XG(NS),YG(NS),TH,DZA,DZB)
339           ZSTER(NS)=TH
340           DO 30 KK=1,NGAMA
341           UP(NS,KK)=WXU(K,KK)
342           VP(NS,KK)=WYU(K,KK)
343        30 CONTINUE
344        27 CONTINUE
345     C
346           DO 28 J=1,NSSTA
347           IF (SCOX(J).LT.XO.OR.SCOX(J).GT.XO+SLX.OR.SCOY(J).LT.YO.OR.
348          +    SCOY(J).GT.YO+SLY) GO TO 28
349           NS=NS+1
350             XG(NS)=(SCOX(J)-XO)/SLX
351             YG(NS)=(SCOY(J)-YO)/SLY
352             CALL TETRI(XG(NS),YG(NS),TH,DZA,DZB)
353             ZSTER(NS)=TH
354             DO 28 KK=1,NGAMA
355               UP(NS,KK)=WIX(J,KK)
356               VP(NS,KK)=WIY(J,KK)
357        28 CONTINUE
358     C
359           IF(NOSTA.EQ.0) GO TO 29
360           DO 29 J=1,NOSTA
361             IF (OCOX(J).LT.XO.OR.OCOX(J).GT.XO+SLX.OR.OCOY(J).LT.YO.OR.
362          +      OCOY(J).GT.YO+SLY) GO TO 28
363             NS=NS+1
364             XG(NS)=(OCOX(J)-XO)/SLX
365             YG(NS)=(OCOY(J)-YO)/SLY
366             CALL TETRI(XG(NS),YG(NS),TH,DZA,DZB)
367             ZSTER(NS)=TH
368             DO 29 KK=1,NGAMA
369               UP(NS,KK)=OWIX(J,KK)
370               VP(NS,KK)=OWIY(J,KK)
371        29 CONTINUE
372     C
```

```
C              ***AMPL DETERMINES A LEAST-SQUARES SOLUTION FOR FI, GI,
C              ***AND GMF.
      NTERMS=NTEMAX
      CALL AMPL (NTERMS,NS,NGAMA,HTGAMA,HA,ROA,HH,IHH,RXY)
C
C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
      IF(DETAIL) THEN
        WRITE(LUO,45) NTERMS
        DO 55 K=1,NS
C            **COMPUTE OUTPUT WINDS ON ALL GAMMA SURFACES
          CALL TETRI(XG(K),YG(K),TH,DZA,DZB)
          CALL WINGAM(XG(K),YG(K),TH,DZA,DZB,NTERMS,NGAMA,HTGAMA,
     +                HA,ROA,HH,IHH,SLX,RXY,WX,WY,WZ)
          DO 54 KK=1,NGAMA
            CALL WNDRSP(UP(K,KK),VP(K,KK),WDI(KK),WSI(KK))
            CALL WNDRSP(WX(KK),WY(KK),WDO(KK),WSO(KK))
   54     CONTINUE
          GRX=XG(K)*SLX+REAL(IX0)
          GRY=YG(K)*SLY+REAL(IY0)
          WRITE(LUO,46) GRX,GRY,(WDI(KK),WSI(KK),KK=1,NGAMA)
          WRITE(LUO,47) GRX,GRY,(WDO(KK),WSO(KK),KK=1,NGAMA)
   55   CONTINUE
      ENDIF
C*******************************************************************************
C
C              ***FROUDE ADJUSTS WIND FIELD ACCORDING TO FROUDE NUMBERS.
      IF (FROU)  THEN
        IHRR=IHR
        CALL FROUDE (JUL,IHRR,NS,NGAMA,HTGAMA,NTERMS,NTERMA,RAD,NSTA,
     +               NSSTA,IBTZ,FRO1ST,HA,ROA,NSA,LUO)
C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
        IF(DETAIL) THEN
         IF(NTERMA.NE.0) THEN
          WRITE(LUO,49) NTERMA
          DO 65 K=1,NS
C            **COMPUTE OUTPUT WINDS ON ALL GAMMA SURFACES
            CALL TETRI(XG(K),YG(K),TH,DZA,DZB)
            CALL WINGAM(XG(K),YG(K),TH,DZA,DZB,NTERMA,NGAMA,HTGAMA,
     +                  HA,ROA,HH,IHH,SLX,RXY,WX,WY,WZ)
            DO 64 KK=1,NGAMA
              CALL WNDRSP(UP(K,KK),VP(K,KK),WDI(KK),WSI(KK))
              CALL WNDRSP(WX(KK),WY(KK),WDO(KK),WSO(KK))
   64       CONTINUE
            GRX=XG(K)*SLX+REAL(IX0)
            GRY=YG(K)*SLY+REAL(IY0)
            WRITE(LUO,46) GRX,GRY,(WDI(KK),WSI(KK),KK=1,NGAMA)
            WRITE(LUO,47) GRX,GRY,(WDO(KK),WSO(KK),KK=1,NGAMA)
   65     CONTINUE
          ENDIF
        ENDIF
C*******************************************************************************
      ELSE
        NTERMA=0
      ENDIF
      NTE=NTERMS
      IF (NTERMA.NE.0) NTE=NTERMA
C
C              ***WRITE OUT AMPLITUDE FUNCTIONS FOR THIS TIME STEP.
      DO 32 J=1,NGAMA
      WRITE (12) ICLK,THETTA,NTE,(FI(K,J),GI(K,J),GMF(K,J),K=1,NTE)
   32 CONTINUE
C
```

```
C               ***END OF TIME LOOP.
  500 CONTINUE
C
      RETURN
   35 FORMAT (I6,1X,A10,2F10.3,I4)
   36 FORMAT (' UPPER AIR STATION ID AND SURFACE STATION ID DO NOT',' MA
     +TCH.')
   40 FORMAT(1H1,///,5X,'TRACE OF SUBROUTINE AMPLIT EXECUTION - AMPLIT',
     +' COMPUTES THE AMPLITUDE FUNCTIONS FOR EACH HOUR FOR EACH "GAMMA"'
     +,' SURFACE',/,5X,'USING THE HOURLY INPUT WINDS FOR EACH "GAMMA"',
     +' SURFACE.  THE BASE TIME ZONE IS',A10,'.' )
   41 FORMAT(1H0,/,5X,'THE WINDS ARE COMPUTED ON THE FOLLOWING "GAMMA"',
     +' SURFACES :',/,7X,'GAMMA =',11(2X,F6.4))
   48 FORMAT(1H ,4X,'THE UPPER BOUNDARY IS AT A CONSTANT',F6.0,
     +' M AGL.')
   50 FORMAT(1H ,4X,'THE UPPER BOUNDARY IS AT A CONSTANT',F6.0,
     +' M MSL.')
   39 FORMAT(1H ,4X,'THE NUMBER OF POLYNOMIAL TERMS ARE ',I2,' .')
   42 FORMAT(1H0,/,5X,'SUMMARY OF OTHER SPECIFIED POINTS WHERE THE',
     +' INPUT WINDS ARE INTERPOLATED TO:',/,10X,'NAME',6X,'ID NO',2X,
     +'X (KM)',2X,'Y (KM)')
   43 FORMAT(1H ,6X,A10,2X,I6,3X,F5.1,3X,F5.1)
   44 FORMAT(1H0,/,5X,'JULIAN DAY ='1X,I3,5X,'HOUR ='1X,I2,10X,
     +'(OUTPUT FROM SUBROUTINE AMPLIT)')
   45 FORMAT(1H ,/,7X,'THE INPUT WINDS AND COMPUTED WINDS AT THE',
     +' INPUT POINTS ARE ( NTERMS =',I3,' ):',/,31X,'SURFACE 1',2X,
     +'SURFACE 2',2X,'SURFACE 3',2X,'SURFACE 4',2X,'SURFACE 5',2X,
     +'SURFACE 6',2X,'SURFACE 7',2X,'SURFACE 8',2X,'SURFACE 9',/,9X,
     +'I/O',2X,'X (KM)',2X,'Y (KM)',9(4X,'WD',3X,'WS'))
   46 FORMAT(1H ,9X,'I',4X,F4.0,4X,F4.0,2X,9(2X,F4.0,1X,F4.1))
   47 FORMAT(1H ,9X,'O',4X,F4.0,4X,F4.0,2X,9(2X,F4.0,1X,F4.1))
   49 FORMAT(1H ,/,7X,'THE INPUT WINDS AND COMPUTED WINDS AT THE INPUT',
     +' POINTS AFTER FROUDE MODS ARE ( NTERMS =',I3,' ):',/,31X,
     +'SURFACE 1',2X,'SURFACE 2',2X,'SURFACE 3',2X,'SURFACE 4',2X,
     +'SURFACE 5',2X,'SURFACE 6',2X,'SURFACE 7',2X,'SURFACE 8',2X,
     +'SURFACE 9',/,9X,'I/O',2X,'X (KM)',2X,'Y (KM)',9(4X,'WD',3X,'WS'))
      END
```

```
001     C  SUBROUTINE: MET.BETA
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO COMPUTE THE WARMING RATE ABOVE A
005     C  VALLEY ATMOSPHERE DURING THE DAY.  THE WARMING RATE (DEG K/S) IS
006     C  NEEDED IN THE COMPUTATION OF VALLEY TEMPERATURE INVERSION
007     C  DESTRUCTION DURING THE MORNING TRANSITION PERIOD.  THE WARMING
008     C  RATE IS COMPUTED AS THE DIFFERENCE IN POTENTIAL TEMPERATURE
009     C  BETWEEN TWO SOUNDINGS AT THE VALLEY TOP, DIVIDED BY THE TIME
010     C  BETWEEN THE TWO SOUNDINGS.
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE INPUTS ARE:
013     C       NOBM = NUMBER OF OBS. IN EARLIER SOUNDING
014     C       NOBA = NUMBER OF OBS IN LATER SOUNDING
015     C       HTOM = ARRAY OF OBS. HEIGHTS FOR EARLIER SOUNDING (M. MSL)
016     C       HTOA = ARRAY OF OBS. HEIGHTS FOR LATER SOUNDING (M. MSL)
017     C       PTOM = ARRAY OF POT. TEMP. OBS. FOR EARLIER SND. (DEG K)
018     C       PTOA = ARRAY OF POT. TEMP. OBS. FOR LATER SND. (DEG K)
019     C       IHRM = HOUR OF DAY OF EARLIER SOUNDING (MST)
020     C       IHRA = HOUR OF DAY OF LATER SOUNDING (MST)
021     C       ELVAL = AVERAGE ELEVATION OF VALLEY FLOOR (M. MSL)
022     C       DPVAL = AVERAGE DEPTH OF VALLEY (M)
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  THE OUTPUTS ARE:
025     C       BTA = WARMING RATE ABOVE VALLEY ATMOSPHERE (DEG K/S)
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLING SUBROUTINES:
028     C       COEFF
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  CALLED SUBROUTINES:
031     C       TPROF
032     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
033     C  COMMON BLOCKS:
034     C       (NONE)
035     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
036     C  PARAMETER DEFINITIONS:
037     C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
038     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
039     C
040           SUBROUTINE BETA(NOBM,NOBA,HTOM,HTOA,PTOM,PTOA,IHRM,
041          +IHRA,ELVAL,DPVAL,BTA)
042     C
043           PARAMETER NOB=30
044     C
045           DIMENSION HTOM(NOB),HTOA(NOB),PTOM(NOB),PTOA(NOB)
046     C
047           IF(ELVAL.GT.HTOM(1)) THEN
048             HT=ELVAL+DPVAL
049           ELSE
050             HT=HTOM(1)+DPVAL
051           ENDIF
052     C
053           CALL TPROF (NOBM,HTOM,PTOM,HT,VALM)
054           CALL TPROF (NOBA,HTOA,PTOA,HT,VALA)
055           BTA=(1./REAL(IHRA-IHRM))*(VALA-VALM)/3600.
056           IF(BTA.LT.0.) BTA=0.
057     C
058           RETURN
059           END
```

```
001     C  SUBROUTINE: MET.BLPARS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE GRIDDED FRICTION VELOCITY,
005     C  CONVECTIVE VELOCITY, MONIN-OBUKOV LENGTH, PRESSURE AND TEMPERATURE
006     C  FOR EACH PROCESSING TIME STEP.  INPUT DATA IS FROM SURFACE
007     C  OBSERVATIONS.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C    THRU ARGUMENT:
011     C      IJS = STARTING JULIAN DATE
012     C      IJE = ENDING JULIAN DATE
013     C      IHS = STARTING HOUR-OF-DAY
014     C      IHE = ENDING HOUR-OF-DAY
015     C      RAD = MAXIMUM SEPARATION DISTANCE, BETWEEN A SURFACE OR 'OTHER'
016     C            STATION AND AN UPPER AIR STATION, AT WHICH FOR ANY ACTUAL
017     C            SEPARATION DISTANCE GREATER THAN 'RAD', A 1/R**2
018     C            INTERPOLATION OF THE SOUNDING TO THE SURFACE OR 'OTHER'
019     C            STATION WILL NOT BE PERFORMED (KM).
020     C      NGAMA = NUMBER OF 'GAMMA' SURFACES
021     C      HTGAMA = ARRAY OF HEIGHT OF GAMMA SURFACES (GAMMA COORD)
022     C      NSTA = NUMBER OF UPPER AIR STATIONS
023     C      NSSTA = NUMBER OF SURFACE STATIONS
024     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
025     C                              8(PACIFIC)
026     C      TZNAM = BASE TIME ZONE NAME
027     C      DETAIL = LOGICAL VARIABLE IF .TRUE. THEN WRITE OUT INFORMATION
028     C               TO UNIT LUO
029     C      ZNOT = DEFAULT SURFACE ROUGHNESS VALUE, OR IF EQUAL TO '999.'
030     C             THEN DETERMINE ZO FROM LAND USE CATEGORY
031     C      LUO = LOGICAL UNIT TO WHICH PRINTOUT IS DIRECTED
032     C    THRU COMMON BLOCK 'MAIN2':
033     C      IHH = 1  'SMALL H' = CONSTANT
034     C          = 2  'BIG H' = CONSTANT
035     C          = 3  'SMALL H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
036     C          = 4  'BIG H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
037     C      HH = VALUE FOR 'SMALL H' OR 'BIG H' DEPENDING ON WHETHER
038     C           IHH=1 OR IHH=2, RESPECTIVELY (M)
039     C      SLX = DOMAIN WIDTH IN X-DIR (KM)
040     C      SLY = DOMAIN WIDTH IN Y-DIR (KM)
041     C      RXY = SLX/SLY
042     C    THRU COMMON BLOCK 'SRFC1':
043     C      IDS = ARRAY OF SURFACE STATIONS ID NUMBERS
044     C      ELS = ELEVATION OF SURFACE STATION (M MSL)
045     C      NAMES = NAME OF SURFACE STATION
046     C      NOBS = NUMBER OF SURFACE OBSERVATIONS FOR SURFACE STATION
047     C      SCOX = X-COORD IN GRAMA SYSTEM OF SURFACE STATION (KM)
048     C      SCOY = Y-COORD IN GRAMA SYSTEM OF SURFACE STATION (KM)
049     C      SLAT = LATITUDE OF SURFACE STATION (DEG)
050     C      SLONG = LONGITUDE OF SURFACE STATION (DEG)
051     C      ISSRT = ARRAY OF STARTING RECORD NUMBER OF SRFC OBS FOR ALL
052     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
053     C      ISEND = ARRAY OF ENDING RECORD NUMBER OF SRFC OBS FOR ALL
054     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
055     C    THRU LOGICAL UNIT 12:
056     C      FI = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR EACH 'GAMMA' SRFC
057     C      GI = ARRAY OF AMPLITUDE FUNCTION 'GSUGI' FOR EACH 'GAMMA' SRFC
058     C      GMF = ARRAY OF AMPLITUDE FUNCTION 'GSUBI-FSUBI' FOR EACH 'GAMMA'
059     C            SURFACE
060     C    THRU LOGICAL UNIT 17:
061     C      HMD = MIXING HEIGHT (M)
062     C    THRU LOGICAL UNIT 25:
```

```
063     C       LANDU = LAND USE CATEGORY
064     C     THRU COMMON BLOCK 'GRID1':
065     C       IXOS = X-COORD ORIGIN OF BOUNDARY LAYER PARAMETERS GRID (KM)
066     C       IYOS = Y-COORD ORIGIN OF BOUNDARY LAYER PARAMETERS GRID (KM)
067     C       ICSS = CELL SIZE OF BOUNDARY LAYER PARAMETERS GRID (KM)
068     C       NXCS = NUMBER OF GRID CELLS IN X-DIR
069     C       NYCS = NUMBER OF GRID CELLS IN Y-DIR
070     C       XR = ARRAY OF GRID POINTS X-COORDINATES (KM)
071     C       YR = ARRAY OF GRID POINTS Y-COORDINATES (KM)
072     C       A = ARRAY OF GRID POINTS ALPHA COORDINATES
073     C       B = ARRAY OF GRID POINTS BETA COORDINATES
074     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
075     C  THE OUTPUTS ARE:
076     C     THRU COMMON BLOCK 'AMPLFN':
077     C       FI = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR EACH 'GAMMA' SRFC
078     C       GI = ARRAY OF AMPLITUDE FUNCTION 'GSUGI' FOR EACH 'GAMMA' SRFC
079     C       GMF = ARRAY OF AMPLITUDE FUNCTION 'GSUBI-FSUBI' FOR EACH 'GAMMA'
080     C             SURFACE
081     C     THRU LOGICAL UNIT 20:
082     C       USTR = ARRAY OF GRIDDED FRICTION VELOCITY FOR TIME STEP
083     C       CAPL = ARRAY OF GRIDDED MONIN-OBUKOV LENGTHS FOR TIME STEP
084     C       WSTR = ARRAY OF GRIDDED CONVECTIVE VELOCITIES FOR TIME STEP
085     C       TSURFA = ARRAY OF GRIDDED SURFACE TEMPERATURES FOR TIME STEP
086     C       PRESS = ARRAY OF GRIDDED SURFACE PRESSURES FOR TIME STEP
087     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
088     C  FILES READ FROM:
089     C       LOGICAL UNIT 12 = UNFORMATTED SEQUENTIAL DISK FILE CONTAINING
090     C                         AMPLITUDE FUNCTIONS FOR EACH 'GAMMA' SRFC
091     C       LOGICAL UNIT 17 = UNFORMATTED SEQUENTIAL DISK FILE CONTAINING
092     C                         GRIDDED MIXING HEIGHTS
093     C       LOGICAL UNIT 25 = FORMATTED SEQUENTIAL DISK FILE CONTAINING
094     C                         GRIDDED LAND USE CATEGORIES
095     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
096     C  FILES WRITTEN TO:
097     C       LOGICAL UNIT 20 = UNFORMATTED SEQUENTIAL DISK FILE CONTAINING
098     C                         GRIDDED BOUNDARY LAYER PARAMETERS
099     C       LOGICAL UNIT LUO = IF DETAIL IS .TRUE. THEN RUN INFO IS OUTPUT
100     C       LOGICAL UNIT  6 = ERROR MESSAGES
101     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
102     C  CALLING SUBROUTINES:
103     C       AAMAIN
104     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
105     C  CALLED SUBROUTINES:
106     C       GRID, MONIN, READS, SOLAR, TETRI, USTAR, WINDS, WSTAR, SUBA0
107     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
108     C  COMMON BLOCKS:
109     C       MAIN2, SRFC1, AMPLFN, GRID1
110     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
111     C  PARAMETERS DEFINITIONS:
112     C       NGA = NUMBER OF 'GAMMA' SURFACES
113     C       NNS = NUMBER OF SURFACE STATIONS
114     C       NUA = NUMBER OF UPPER AIR STATIONS
115     C       NPO = NUMBER OF POLYNOMIAL TERMS
116     C       NPG = NUMBER OF GRID POINTS IN X-DIR OR Y-DIR FOR BOUNDARY
117     C             LAYER PARAMETERS GRID
118     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
119     C
120           SUBROUTINE BLPARS(IJS,IJE,IHS,IHE,NSTA,NSSTA,NGAMA,HTGAMA,RAD,
121          +                  IBTZ,TZNAM,DETAIL,ZNOT,LUO)
122     C
123           PARAMETER
124          +    NGA=9,          NNS=15,         NPO=21,         NPG=10,
```

```
125          +    NUA=10
126     C
127           LOGICAL
128          +    DETAIL,          PASS,          RDS1ST,          A01ST
129     C
130           CHARACTER*10
131          +    NAMES,           TZNAM
132     C
133           DIMENSION
134          +    BETA(11),        CAPL(NPG,NPG), CPT(12),         HMD(NPG),
135          +    HTGAMA(NGA),     ICC(NNS),      IND(NNS),        JDAE(12),
136          +    LANDU(NPG),      NOBSS(NNS),    SLP(12),         TSURF(NNS),
137          +    USTR(NPG,NPG),   WSTR(NPG,NPG), XLANDU(12),      Z0(NPG,NPG),
138          +    TSURFA(NPG,NPG),PRESS(NPG,NPG), A0U(NUA),        A0S(NNS)
139     C
140           COMMON /MAIN2/
141          +    HH,              IHH,           IX0,             IY0,
142          +    RXY,             SLX,           SLY
143     C
144           COMMON /AMPLFN/
145          +    FI(NP0,NGA),     GI(NP0,NGA),   GMF(NP0,NGA)
146     C
147           COMMON /SRFC1/
148          +    ELS(NNS),        IDS(NNS),      NAMES(NNS),      NOBS(NNS),
149          +    SCOX(NNS),       SCOY(NNS),     SLAT(NNS),       SLONG(NNS),
150          +    ISSRT(NNS),      ISEND(NNS)
151     C
152           COMMON /GRID1/
153          +    IX0S,            IY0S,          NXCS,            NYCS,
154          +    ICSS,            A(NPG),        B(NPG),          XR(NPG),
155          +    YR(NPG)
156     C
157     C                SPECIFY CONSTANTS
158           ZMS=10.
159           RDS1ST=.TRUE.
160           A01ST=.TRUE.
161     C
162     C                BETA IS THE SOLAR RADIATION REDUCTION FACTOR
163           DATA BETA/1.0,.91,.84,.79,.75,.72,.68..62,.53,.41,.23/
164     C
165     C                CONSTANTS USED IN TEMPERATURE CORRECTION FOR ALTITUDE
166     C                DIFFERENCES.  BASED ON MONTHLY AVERAGE LAPSE RATES
167     C                COMPUTED FROM SURFACE TEMPERATURES IN WESTERN COLORADO.
168           DATA JDAE/16,75,105,136,166,197,228,258,289,319,350,366/
169           DATA SLP/-.0152,-.0607,.0093,-.0552,.0180,-.0135,.0116,.0037,
170          +          .0832,.0500,.0261,-.0152/
171           DATA CPT/-1.6950,-.9592,-6.2100,.5619,-9.3880,-4.1510,-9.1077,
172          +          -7.2960,-27.8223,-18.2200,-10.6052,3.8600/
173     C
174     C                SPECIFY Z0 FOR EACH LAND USE CATEGORY
175           DATA XLANDU/.2,.3,.05,.9,1.,.1,.2,.3,.2,.5,1.,.0001/
176     C
177     C                ***DETERMINE SURFACE ROUGHNESS (Z0) FROM GRIDDED LAND USE
178     C                ***CATEGORIES.
179           IF(ZNOT.EQ.999.) THEN
180             DO 103 J=1,NYCS
181             READ(25,5) (LANDU(I),I=1,NXCS)
182             DO 103 I=1,NXCS
183             Z0(I,J)=XLANDU(LANDU(I))
184      103    CONTINUE
185           ELSE
186             DO 102 J=1,NYCS
```

```
187             DO 102 I=1,NXCS
188             Z0(I,J)=ZNOT
189         102 CONTINUE
190           ENDIF
191     C
192     C               ***REWIND AMPLITUDE FILE.
193           REWIND 12
194           READ (12) ISKP
195           READ (12) ISKP
196     C
197     C               ***REWIND MIXING HEIGHT FILE.
198           REWIND 17
199           READ (17) ISKP
200     C
201     C               ***WRITE FILE ID'S AND DOMAIN SPECS TO UNIT 20.
202           WRITE (20) IXOS,IYOS,NXCS,NYCS,ICSS
203     C
204     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
205           WRITE(LUO,60) TZNAM
206           WRITE(LUO,20) IXOS,IYOS,NXCS,NYCS,ICSS
207           WRITE(LUO,21) (XR(I),I=1,NXCS)
208           DO 22 J=NYCS,1,-1
209             WRITE(LUO,23) YR(J),(Z0(I,J),I=1,NXCS)
210        22 CONTINUE
211     C*******************************************************************************
212     C
213     C               ***LOOP ON JULIAN DAY.
214           DO 100 JUL=IJS,IJE
215     C
216     C               ***COMPUTE SURFACE TEMPERATURE CORRECTION LAPSE RATE IN
217     C               ***DEGF/1000FT
218           DO 98 LAPSC=1,12
219           IF(JUL.LE.JDAE(LAPSC)) GO TO 99
220        98 CONTINUE
221        99 TLR=SLP(LAPSC)*JUL+CPT(LAPSC)
222     C
223     C               ***COMPUTE A0 AT EACH SURFACE STATION
224           IA0=2
225           CALL SUBA0(JUL,NSTA,NSSTA,IBTZ,RAD,A01ST,IA0,A0U.A0S)
226           A01ST=.FALSE.
227     C
228     C               ***LOOP ON HOUR OF THE DAY
229           ISA=1
230           ISP=24
231           IF (JUL.EQ.IJS) ISA=IHS
232           IF (JUL.EQ.IJE) ISP=IHE
233           ICLKK=(JUL-1)*24
234           DO 30 N=ISA,ISP
235     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
236           IF(DETAIL) THEN
237             WRITE(LUO,63) JUL,N
238           ENDIF
239     C*******************************************************************************
240           HR=FLOAT(N)
241           ICLK=ICLKK+N
242     C               ***READ IN AMPLITUDE FUNCTIONS FOR HOUR AND ALSO
243     C               ***COMPUTE CONSTANTS USED IN DENSITY CALCULATION
244           DO 26 J=1,NGAMA
245           READ (12) ICK,THETTA,NTE,(FI(K,J),GI(K,J),GMF(K,J),K=1,NTE)
246        26 CONTINUE
247           IF (ICK.NE.ICLK)  THEN
248             WRITE (6,39)
```

```
249           STOP 'BLPARS'
250           ENDIF
251           ROA=353.1/THETTA
252           HA=102.4*THETTA
253     C
254     C                ***IDENTIFY SURFACE STATIONS TO USE FOR GIVEN HOUR.
255           NS=0
256           DO 24 K=1,NSSTA
257           CALL READS (ICLK,NOBSS(K),ISSRT(K),NOBS(K),K,RDS1ST,PASS,BL,
258          +            BL,TSURFI,BL,ICCI)
259           IF (PASS) GO TO 24
260           IF (AOS(K).EQ.999) GO TO 24
261           NS=NS+1
262           TSURF(NS)=TSURFI
263           ICC(NS)=ICCI
264           IND(NS)=K
265        24 CONTINUE
266           RDS1ST=.FALSE.
267     C
268           IF(NS.EQ.0) THEN
269             WRITE(6,50)JUL,N
270             STOP 'BLPARS'
271           ENDIF
272     C
273     C          LOOP ON GRID POINTS
274           DO 29 J=1,NYCS
275           READ(17) ICCK,(HMD(JJ),JJ=1,NXCS)
276           READ(17) ISKP
277           DO 28 I=1,NXCS
278     C
279     C                **DETERMINE HEIGHT OF GRID POINT (1000FT)
280     C                **304.8 CONVERTS METERS TO THOUSANDS OF FEET
281           CALL TETRI(A(I),B(J),THM,DZA,DZB)
282           TH=THM/304.8
283     C
284     C                **1/R**2 WEIGHT BOUNDARY LAYER PARAMETERS
285           WF=0.
286           AOSS=0.
287           TSURFS=0.
288           CCS=0.
289           DO 27 K=1,NS
290           RDIF=SQRT((SCOX(IND(K))-XR(I))**2.+(SCOY(IND(K))-YR(J))**2.)
291           IF (RDIF.LE.RAD)  THEN
292           IF (RDIF.LE..1) RDIF=1.*10.**(-3.)
293           REC=1./RDIF**2
294           WF=REC+WF
295     C                **DETERMINE ADJUSTED TEMPERATURE (DEG C) DUE TO ELEV.
296     C                ** DIFFERENCE BETWEEN SURFACE STATION AND GRID POINT.
297     C                **(1.8 CONVERTS DEGF TO DEGC)
298           TCOR=TSURF(K)+(TH-ELS(IND(K))/304.8)*TLR/1.8
299           TSURFS=REC*TCOR+TSURFS
300           CCS=REC*FLOAT(ICC(K))+CCS
301           AOSS=REC*AOS(IND(K))+AOSS
302           ENDIF
303        27 CONTINUE
304           TSURFA(I,J)=TSURFS/WF+273.15
305           CCD=CCS/WF
306           ICCD=NINT(CCD)
307           AOSD=AOSS/WF
308     C
309           CALL WINDS (A(I),B(J),10.,NTE,NGAMA,HTGAMA,HA,ROA,HH,IHH,SLX,RXY,
310          +            U,V,W,WT)
```

```
311           WS=SQRT(U**2+V**2)
312           CALL GRID(XR(I),YR(J),XLATG,YLONGG)
313      C                 **COMPUTE SOLAR CONSTANT
314           TSN=30.
315           CALL SOLAR(JUL,TSN,XLATG,YLONGG,IBTZ,A1,ALPHA,TSR,TSS)
316           SOLC=A1/ALPHA
317      C                 **COMPUTE INCOMMING SOLAR RADIATION
318           CALL SOLAR(JUL,HR,XLATG,YLONGG,IBTZ,A1,ALPHA,TSR,TSS)
319           SRADD=SOLC*BETA(ICCD+1)*ALPHA
320           HOD=-25.5+2.4*CCD
321           RHOD=ROA*(1.-THM/HA)**2.5
322           WT=(AOSD*SRADD+HOD)/(996.*RHOD)
323           CALL USTAR(TSURFA(I,J),WS,ZMS,Z0(I,J),WT,USTR(I,J))
324           CALL MONIN(TSURFA(I,J),WT,USTR(I,J),CAPL(I,J))
325           CALL WSTAR(TSURFA(I,J),WT,HMD(I),WSTR(I,J))
326      C                 *2.869 IS THE UNIVERSAL GAS CONST [(M**3-MB)/(KG-DEGK)]
327           PRESS(I,J)=2.869*RHOD*TSURFA(I,J)
328      C
329        28 CONTINUE
330           WRITE (20) ICLK,(USTR(I,J),I=1,NXCS)
331           WRITE (20) ICLK,(CAPL(I,J),I=1,NXCS)
332           WRITE (20) ICLK,(WSTR(I,J),I=1,NXCS)
333           WRITE (20) ICLK,(TSURFA(I,J),I=1,NXCS)
334           WRITE (20) ICLK,(PRESS(I,J),I=1,NXCS)
335      C
336      C                  ***END OF GRID LOOP
337        29 CONTINUE
338      C
339      C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
340           IF(DETAIL) THEN
341             WRITE(LUO,64) (XR(I),I=1,NXCS)
342             DO 31 J=NYCS,1,-1
343               WRITE(LUO,65) YR(J),(TSURFA(I,J),I=1,NXCS)
344        31   CONTINUE
345             WRITE(LUO,66) (XR(I),I=1,NXCS)
346             DO 32 J=NYCS,1,-1
347               WRITE(LUO,65) YR(J),(PRESS(I,J),I=1,NXCS)
348        32   CONTINUE
349             WRITE(LUO,67) (XR(I),I=1,NXCS)
350             DO 33 J=NYCS,1,-1
351               WRITE(LUO,68) YR(J),(USTR(I,J),I=1,NXCS)
352        33   CONTINUE
353             WRITE(LUO,70) (XR(I),I=1,NXCS)
354             DO 34 J=NYCS,1,-1
355               WRITE(LUO,71) YR(J),(CAPL(I,J),I=1,NXCS)
356        34   CONTINUE
357             WRITE(LUO,69) (XR(I),I=1,NXCS)
358             DO 35 J=NYCS,1,-1
359               WRITE(LUO,68) YR(J),(WSTR(I,J),I=1,NXCS)
360        35   CONTINUE
361           ENDIF
362      C*******************************************************************************
363      C
364      C                  ***END OF HOUR LOOP
365        30 CONTINUE
366      C                  ***END OF DAY LOOP
367       100 CONTINUE
368      C
369           RETURN
370         5 FORMAT (10I3)
371        39 FORMAT (2X,'AMPLITUDE FILE CLOCK DOES NOT MATCH BLPARS CLOCK.')
372        50 FORMAT (2X,'NO SURFACE STATIONS FOR HOUR',I3,' ON DAY',I4,' .')
```

```
373         60 FORMAT(1H1,///,5X,'TRACE OF SUBROUTINE BLPARS EXECUTION - BLPARS',
374           +' COMPUTES HOURLY GRIDDED BOUNDARY LAYER PARAMETERS FROM SURFACE',
375           +/,5X,'WEATHER OBSERVATIONS.  THE BASE TIME ZONE IS',A10,'.' )
376         20 FORMAT(1H0,/,5X,'THE GRID SPECIFICATIONS ARE:',/,7X,'IXOS = ',I3,
377           +5X,'IYOS = ',I3,5X,'NXCS = ',I2,6X,'NYCS = ',I2,6X,'ICSS = ',I3)
378         21 FORMAT(1H0,/,5X,'THE INPUT SURFACE ROUGHNESS IS (M):',/,
379           +43X,'X (KM)',/,7X,'Y (KM)',1X,10(2X,F5.0),/)
380         23 FORMAT(1H ,7X,F4.0,2X,10(2X,F5.2))
381         63 FORMAT(1H0,/,5X,'JULIAN DAY ='1X,I3,5X,'HOUR ='1X,I2,10X,
382           +'(OUTPUT FROM SUBROUTINE BLPARS)')
383         64 FORMAT(1H ,/,7X,'THE SURFACE TEMPERATURES ARE (DEG K):',/,43X,
384           +'X (KM)',/,9X,'Y (KM)',1X,10(2X,F7.0),/)
385         65 FORMAT(1H ,9X,F4.0,2X,10(2X,F7.0))
386         66 FORMAT(1H ,/,7X,'THE SURFACE PRESSURES ARE (MB):',/,43X,'X (KM)',
387           +/,9X,'Y (KM)',1X,10(2X,F7.0),/)
388         67 FORMAT(1H ,/,7X,'THE FRICTION VELOCITIES ARE (M/S):',/,43X,
389           +'X (KM)',/,9X,'Y (KM)',1X,10(2X,F7.0),/)
390         68 FORMAT(1H ,9X,F4.0,2X,10(2X,F7.3))
391         69 FORMAT(1H ,/,7X,'THE CONVECTIVE VELOCITIES ARE (M/S):',/,43X,
392           +'X (KM)',/,9X,'Y (KM)',1X,10(2X,F7.0),/)
393         70 FORMAT(1H ,/,7X,'THE MONIN-OBUKOV LENGTHS ARE (M):',/,43X,
394           +'X (KM)',/,9X,'Y (KM)',1X,10(2X,F7.0),/)
395         71 FORMAT(1H ,9X,F4.0,2X,10(2X,F7.1))
396            END
```

```
001     C  SUBROUTINE: MET.COEFF
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE COMPUTES THE COUPLING COEFFICIENTS FOR EACH
005     C  VALLEY SEGMENT FOR EACH HOURLY TIME STEP DURING A PROCESSING
006     C  PERIOD.  THE COUPLING COEFFICIENT AT EACH VALLEY SEGMENT IS
007     C  COMPUTED USING THE NEAREST TEMPERATURE SOUNDING TO THE VALLEY
008     C  CENTER AND THE PHYSICAL CHARACTERISTICS OF EACH VALLEY SEGMENT.
009     C  A COUPLING COEFFICIENT OF '1' MEANS THE VALLEY ATMOSPHERE IS
010     C  COUPLED WITH THE FLOW ABOVE RIDGETOPS, AND A COEFFICIENT OF '0'
011     C  MEANS THE FLOW IS DECOUPLED.  THE COUPLING COEFFICIENT CAN GO FROM
012     C  '1' TO '0' IN ONE TIME STEP BUT CANNOT GO FROM '0' (DECOUPLED) TO
013     C  '1' (COUPLED) IN ONE TIME STEP.  A COEFFICIENT GOING FROM '0' TO
014     C  '1' NEEDS TO ACCOUNT FOR THE RATE OF RELEASE OF POLLUTANTS FROM
015     C  THE VALLEY ATMOSPHERE TO THE FLOW ABOVE RIDGETOPS.  THIS IS DONE
016     C  DURING THE MORNING TRANSITION PERIOD BY COMPUTING THE VALLEY
017     C  CROSS-SECTIONAL AREA WHICH HAS BEEN CONSUMED BY THE CBL GROWTH AND
018     C  INVERSION DESCENT PER TIME STEP.  THE RATIO OF THIS CONSUMED AREA
019     C  TO THE TOTAL AREA IS THE COEFFICIENT FOR A TIME STEP.  THE
020     C  COEFFICIENT THEREFORE IS A FRACTION BETWEEN 0 AND 1 FOR EACH TIME
021     C  STEP AFTER COUPLING STARTS UNTIL THE INVERSION IS DESTROYED.  THE
022     C  SUM OF THE FRACTIONS TOTALS 1.0 .  IT IS ASSUMED THAT EVEN IF A
023     C  TEMPERATURE INVERSION IS NOT DESTROYED THROUGHOUT THE DAYTIME
024     C  PERIOD,  THE POLLUTANT MASS TRAPPED IN THE VALLEY WOULD BE
025     C  COMPLETELY VENTILATED FROM THE VALLEY BY SUNSET.  THIS IS DONE BY
026     C  EVENLY DISTRIBUTING THE REMAINING POLLUTANT MASS (VALLEY
027     C  CROSS-SECTIONAL AREA) AMONG THE HOURS FROM SUNRISE TO SUNSET.
028     C    A CONSTRAINT ON THE OPERATION OF THIS SUBROUTINE IS THAT THE
029     C  PROCESSING STARTING HOUR BE GREATER THAN OR EQUAL TO SUNRISE AND
030     C  THE STOPPING HOUR BE LESS THAN OR EQUAL TO SUNSET.
031     C    THE VALLEY SEGMENTS MUST BE WITHIN THE COMPUTATIONAL DOMAIN.
032     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
033     C  THE INPUTS ARE:
034     C    THRU ARGUMENT:
035     C      IJS = STARTING JULIAN DATE
036     C      IJE = ENDING JULIAN DATE
037     C      IHS = STARTING HOUR-OF-DAY (1 TO 24)
038     C      IHE = ENDING HOUR-OF-DAY (1 TO 24)
039     C      RAD = MAXIMUM SEPARATION DISTANCE, BETWEEN A SURFACE OR 'OTHER'
040     C            STATION AND AN UPPER AIR STATION, AT WHICH FOR ANY ACTUAL
041     C            SEPARATION DISTANCE GREATER THAN 'RAD', A 1/R**2
042     C            INTERPOLATION OF THE SOUNDING TO THE SURFACE OR 'OTHER'
043     C            STATION WILL NOT BE PERFORMED (KM).
044     C      NSTA = NUMBER OF UPPER AIR STATIONS
045     C      NSSTA = NUMBER OF SURFACE STATIONS
046     C      NVALE = NUMBER OF VALLEYS
047     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
048     C                              8(PACIFIC)
049     C      TZNAM = NAME OF TIME ZONE GIVEN IN PARENS ABOVE.
050     C      DETAIL = LOGICAL VARIABLE IF .TRUE. THEN WRITE OUT INFORMATION
051     C               TO UNIT LUO
052     C      LUO = LOGICAL UNIT TO WHICH THE PRINTOUT IS DIRECTED
053     C    THRU COMMON BLOCK 'UPPER1':
054     C      COX = ARRAY OF X-COORD OF UPPER AIR STATIONS (KM)
055     C      COY = ARRAY OF Y-COORD OF UPPER AIR STATIONS (KM)
056     C      IDU = ARRAY OF UPPER AIR STATIONS IDENTIFICATIONS
057     C      XLAT = LATITUDE OF UPPER AIR STATION (DEG)
058     C      YLONG = LONGITUDE OF UPPER AIR STATION (DEG)
059     C      IUSRTT = ARRAY OF STARTING RECORD NUMBER OF TEMP SNDGS FOR ALL
060     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 9
061     C      IUENDT = ARRAY OF ENDING RECORD NUMBER OF TEMP SNDGS FOR ALL
062     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 9
```

```
063     C       ITZU = TIME ZONE OF STATION = 5(EASTERN), 6(CENTRAL),
064     C                                     7(MOUNTAIN), 8(PACIFIC)
065     C     THRU LOGICAL UNIT 26:
066     C       THE FIRST PART OF THE FILE ON UNIT 26 CONTAINS A RECORD FOR
067     C       EACH VALLEY GIVING THE FOLLOWING INFORMATION:
068     C       IDVAL = ARRAY OF VALLEY ID NUMBERS
069     C       NAMVAL = ARRAY OF VALLEY NAMES
070     C       NSEG = ARRAY OF NUMBER OF SEGMENTS FOR VALLEYS
071     C       XU = X-COORD OF VALLEY CENTER
072     C       YU = Y-COORD OF VALLEY CENTER
073     C       ICODE = CODE SPECIFYING COORDINATE SYSTEM
074     C             =  1 [LATITUDE-LONGITUDE]
075     C             =  2 [UTM ZONE 12]
076     C             =  3 [UTM ZONE 13]
077     C             =  4 [GRAMA]
078     C       VGAMAX = ARRAY OF MAX POT TEMP LAPSE RATE FOR VALLEYS (DEGK/M)
079     C       WSV = ARRAY OF AVERAGE DOWN-VALLEY WIND SPEEDS IN VALLEYS (M/S)
080     C       ELVAL = ARRAY OF VALLEY FLOOR ELEVATIONS (M MSL)
081     C       DPVAL = ARRAY OF VALLEY DEPTHS (M)
082     C       ITZV = TIME ZONE OF VALLEY = 5(EASTERN), 6(CENTRAL),
083     C                                    7(MOUNTAIN), 8(PACIFIC)
084     C       FOLLOWING THE VALLEY HEADER INFORMATION ARE THE INPUT DATA
085     C       FOR EACH VALLEY SEGMENT.  EACH RECORD CONTAINS THE FOLLOWING
086     C       INFORMATION FOR EACH SEGMENT.  THE SEGMENT INFO FOR EACH
087     C       VALLEY IS GROUPED TOGETHER ON THE FILE.
088     C       IDV = VALLEY IDENTIFICATION NUMBER
089     C       IDSEG = SEGMENT IDENTIFICATION NUMBER
090     C       XU = X-COORD OF VALLEY SEGMENT
091     C       YU = Y-COORD OF VALLEY SEGMENT
092     C       ICODE = SAME AS ABOVE
093     C       ELSEG = ELEVATION OF VALLEY SEGMENT FLOOR (M MSL)
094     C       TOSEG = ELEVATION OF RIDGETOPS FOR SEGMENT (M MSL)
095     C       WSEG = WIDTH OF VALLEY SEGMENT FLOOR (M)
096     C       AL1SEG = SLOPE ANGLE OF ONE SEGMENT SIDEWALL (DEG)
097     C       AL2SEG = SLOPE ANGLE OF OTHER SEGMENT SIDEWALL (DEG)
098     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
099     C  THE OUTPUTS ARE:
100     C    THRU LOGICAL UNIT 27:
101     C       IDVAL = ARRAY OF VALLEY ID NUMBERS
102     C       NAMVAL = ARRAY OF VALLEY NAMES
103     C       NSEG = ARRAY OF NUMBER OF SEGMENTS FOR VALLEYS
104     C       COXV = ARRAY OF X-COORD OF VALLEY CENTERS IN GRAMA COORDS (KM)
105     C       COYV = ARRAY OF Y-COORD OF VALLEY CENTERS IN GRAMA COORDS (KM)
106     C       WSV = ARRAY OF AVERAGE DOWN-VALLEY WIND SPEEDS IN VALLEYS (M/S)
107     C       SEGX = ARRAY OF X-COORD OF SEGMENTS IN GRAMA COORDS (KM)
108     C       SEGY = ARRAY OF Y-COORD OF SEGMENTS IN GRAMA COORDS (KM)
109     C       RECORDS ARE WRITTEN FOR EACH HOUR FOR EACH SEGMENT CONTAINING:
110     C       JD = JULIAN DATE
111     C       IHR = HOUR-OF-DAY (1 TO 24)
112     C       IDVAL = ID NUMBER OF VALLEY
113     C       COEF = ARRAY OF SEGMENT COUPLING COEFFICIENT FOR EACH VALLEY
114     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
115     C  CONSTANTS DEFINITIONS:
116     C       DTMO = 3600.  [PROCESSING TIME STEP IN SECONDS]
117     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
118     C  FILES READ FROM:
119     C       LOGICAL UNIT 26 = SEQUENTIAL FORMATTED DISK FILE CONTAINING
120     C                         INPUT DATA ON VALLEYS.
121     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
122     C  FILES WRITTEN TO:
123     C       LOGICAL UNIT 27 = SEQUENTIAL UNFORMATTED DISK FILE CONTAINING
124     C                         VALLEY INFO AND THE COUPLING COEFFICIENTS
```

```
125     C       LOGICAL UNIT LUO = IF DETAIL IS .TRUE. THEN RUN INFO IS OUTPUT
126     C       LOGICAL UNIT  6 = ERROR MESSAGES
127     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
128     C  CALLING SUBROUTINES:
129     C       AAMAIN
130     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
131     C  CALLED SUBROUTINES:
132     C       BETA, CUP, DESCNT, FINDU, GRID, LAT, NEAR, SOLAR, UTM,
133     C       VALGAM, WARM, SUBA0
134     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
135     C  COMMON BLOCKS:
136     C       UPPER1, LAPSE1
137     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
138     C  PARAMETERS DEFINITIONS:
139     C       NUA = NUMBER OF UPPER AIR STATIONS
140     C       NOB = NUMBER OF OBSERVATIONS IN UPPER AIR SOUNDINGS
141     C       NVA = NUMBER OF VALLEYS
142     C       NSG = NUMBER OF SEGMENTS PER VALLEY
143     C       NNS = NUMBER OF SURFACE STATIONS
144     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
145     C
146           SUBROUTINE COEFF(IJS,IJE,IHS,IHE,RAD,NSTA,NSSTA,NVALE,IBTZ,TZNAM,
147          +                 DETAIL,LUO)
148     C
149           PARAMETER
150          +    NSG=10,          NOB=30,          NUA=10,          NVA=5,
151          +    NNS=15
152     C
153           LOGICAL
154          +    DETAIL,          AFT,             MORN,            A01ST
155     C
156           DIMENSION
157          + AL1SEG(NSG,NVA), AL2SEG(NSG,NVA),AOU(NUA),          BL(NOB),
158          + COEFA(NUA),      COEFM(NUA),      COEFM1(NUA),     COXV(NVA),
159          + COYV(NVA),       DPVAL(NVA),      ELSEG(NSG,NVA), ELVAL(NVA),
160          + FRAC(60),        IDVAL(NVA),      IHRA(NUA),       IHRM(NUA),
161          + IS(NOB),         ITZV(NVA),       IUPNT(NUA),      NE(NUA),
162          + NEARS(NUA,NVA),  NSEG(NVA),       PTAX(NOB),       PTMX(NOB),
163          + SEGX(NSG,NVA),   SEGY(NSG,NVA),   THTAX(NOB),      THTAPX(NOB),
164          + THTM1X(NOB),     THTMMX(NOB),     THTMX(NOB),      TMPAX(NOB),
165          + TMPM1X(NOB),     TMPMX(NOB),      TOSEG(NSG,NVA), VGAMAX(NVA),
166          + WSEG(NSG,NVA),   WSV(NVA),        XLATV(NUA),      YLONGV(NUA),
167          + IDSEG(NSG,NVA), COEF1(34,NSG,NVA),  COEF2(34,NSG,NVA),
168          + AOS(NNS)
169     C
170           CHARACTER*10
171          +    NAME,            NAMVAL(NVA),     TZNAM
172     C
173           COMMON /UPPER1/
174          +    COX(NUA),        COY(NUA),        EL(NUA),         IDU(NUA),
175          +    NAME(NUA),       NSDT(NUA),       NSDW(NUA),       XLAT(NUA),
176          +    YLONG(NUA),      IUSRTW(NUA),     IUSRTT(NUA),     IUENDW(NUA),
177          +    IUENDT(NUA),     ITZU(NUA)
178     C
179           COMMON /LAPSE1/
180          +    APT(36,NUA),     ATMP(36,NUA),    DTDZCK(NUA),     ISRS(NUA),
181          +    ISSS(NUA),       NOBA(NUA),       NOBM(NUA),       NOBMM(NUA),
182          +    NOBM1(NUA),      NTS,             PTA(NOB,NUA),    PTM(NOB,NUA),
183          +    PTMM(NOB,NUA),   PTM1(NOB,NUA),   THTA(NOB,NUA),   THTAP(NOB,NUA),
184          +    THTM(NOB,NUA),   THTMM(NOB,NUA),THTM1(NOB,NUA),TMPA(NOB,NUA),
185          +    TMPM(NOB,NUA),   TMPM1(NOB,NUA)
186     C
```

```
C               ***CONSTANTS
      DTMO=3600.
      AC1ST=.TRUE.
C
C               ***INITIALIZE UPPER AIR STATIONS.
      DO 499 L=1,NSTA
        IUPNT(L)=IUSRTT(L)-1
  499 CONTINUE
C
C               ***READ IN VALLEY CHARACTERISTICS.
      DO 4 NV=1,NVALE
        READ(26,360) IDVAL(NV),NAMVAL(NV),NSEG(NV),XU,YU,ICODE,
     +                 VGAMAX(NV),WSV(NV),ELVAL(NV),DPVAL(NV),ITZV(NV)
        GO TO (2,1,1,3), ICODE
    1   IZONE=ICODE
        CALL UTM (IZONE,XU,YU,COXV(NV),COYV(NV))
        CALL GRID (COXV(NV),COYV(NV),XLATV(NV),YLONGV(NV))
        GO TO 4
    2   XLATV(NV)=XU
        YLONGV(NV)=YU
        CALL LAT (XLATV(NV),YLONGV(NV),COXV(NV),COYV(NV))
        GO TO 4
    3   COXV(NV)=XU
        COYV(NV)=YU
        CALL GRID (COXV(NV),COYV(NV),XLATV(NV),YLONGV(NV))
    4 CONTINUE
C
      DO 350 NV=1,NVALE
        DO 345 NSE=1,NSEG(NV)
          READ(26,361) IDV,IDSEG(NSE,NV),XU,YU,ICODE,ELSEG(NSE,NV),
     +    TOSEG(NSE,NV),WSEG(NSE,NV),AL1SEG(NSE,NV),AL2SEG(NSE,NV)
          IF(IDV.NE.IDVAL(NV)) THEN
            WRITE(6,53) IDVAL(NV)
            STOP 'COEFF'
          ENDIF
          GO TO (22,21,21,23), ICODE
   21     IZONE=ICODE
          CALL UTM (IZONE,XU,YU,SEGX(NSE,NV),SEGY(NSE,NV))
          GO TO 345
   22     CALL LAT (XU,YU,SEGX(NSE,NV),SEGY(NSE,NV))
          GO TO 345
   23     SEGX(NSE,NV)=XU
          SEGY(NSE,NV)=YU
  345   CONTINUE
  350 CONTINUE
C
C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
      WRITE(LUO,5) DTMO,TZNAM
      WRITE(LUO,10)
      DO 11 J=1,NVALE
        WRITE(LUO,12) NAMVAL(J),IDVAL(J),COXV(J),COYV(J),XLATV(J),
     +                YLONGV(J),ELVAL(J),DPVAL(J),WSV(J),VGAMAX(J),NSEG(J)
   11 CONTINUE
      DO 13 J=1,NVALE
        WRITE(LUO,14) NAMVAL(J)
      DO 13 I=1,NSEG(J)
        WRITE(LUO,15) IDSEG(I,J),SEGX(I,J),SEGY(I,J),ELSEG(I,J),
     +                TOSEG(I,J),WSEG(I,J),AL1SEG(I,J),AL2SEG(I,J)
   13 CONTINUE
C*******************************************************************************
C
C               ***WRITE OUTPUT FILE ID AND VALLEY INFO TO UNIT 27
```

```
249           WRITE(27) NVALE
250           DO 38 NV=1,NVALE
251             WRITE(27) IDVAL(NV),NAMVAL(NV),NSEG(NV),COXV(NV),COYV(NV),
252          +  WSV(NV),(SEGX(NSE,NV),SEGY(NSE,NV),WSEG(NSE,NV),NSE=1,NSEG(NV))
253        38 CONTINUE
254     C
255     C                ***FIND UPPER AIR STATIONS NEAREST TO EACH VALLEY CENTER.
256           DO 101 NV=1,NVALE
257             CALL NEAR(NSTA,COX,COY,COXV(NV),COYV(NV),RAD,NE)
258             DO 55 NS=1,NSTA
259               NEARS(NS,NV)=NE(NS)
260        55     CONTINUE
261       101 CONTINUE
262     C
263     C                ***INITIALIZE COEFFICIENT ARRAYS TO -999.
264           DO 40 NV=1,NVALE
265           DO 40 NS=1,NSEG(NV)
266           DO 40 NH=1,34
267             COEF1(NH,NS,NV)=-999.
268             COEF2(NH,NS,NV)=-999.
269        40 CONTINUE
270     C
271     C                ***LOOP ON JULIAN DAY
272           DO 900 JD=IJS,IJE
273           ICLKK=(JD-1)*24
274     C
275     C                ***COMPUTE A0 AT EACH UPPER AIR STATION
276           IA0=1
277           CALL SUBA0(JD,NSTA,NSSTA,IBTZ,RAD,A01ST,IA0,A0U,A0S)
278           A01ST=.FALSE.
279     C
280     C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
281           IF(DETAIL) THEN
282             WRITE(LUO,60) JD
283           ENDIF
284     C*******************************************************************************
285     C
286     C                ***LOOP ON VALLEY
287           NACT=0
288           DO 999 NV=1,NVALE
289     C
290     C                ***LOOP ON INDEX OF UPPER AIR STATIONS NEAREST TO
291     C                ***VALLEY
292           DO 500 L=1,NSTA
293           J=NEARS(L,NV)
294           IF(J.EQ.999) GO TO 500
295           IF(A0U(J).EQ.999.) GO TO 500
296     C               *DETERMINE IF ALREADY READ IN DATA FOR THIS STATION
297           IF(NACT.NE.0) THEN
298             DO 44 N=1,NACT
299               IF(J.EQ.IS(N)) GO TO 600
300        44   CONTINUE
301           ENDIF
302     C               *COMPUTE TIME OF SUNRISE AND SUNSET, A1, AND ALPHA
303           TSN=30.
304           CALL SOLAR(JD,TSN,XLAT(J),YLONG(J),ITZU(J),A1,ALPHA,TSR,TSS)
305           ISR=NINT(TSR)+(ITZU(J)-IBTZ)
306           ISS=ISR+NINT(TSS-TSR)
307     C               *FIND MORNING SOUNDING.
308           MORN=.TRUE.
309           AFT=.FALSE.
310           CALL FINDU(JD,ISR,ISS,IDU(J),IUPNT(J),IUENDT(J),MORN,AFT,
```

```
311          +            IHRMX,HM,NOBMX,THTMMX,THTMX,TMPMX,PTMX)
312           IF(.NOT.MORN) GO TO 500
313           CALL CUP(MORN,AFT,NOBMX,THTMX,TMPMX,COEFMX)
314     C              *FIND AFTERNOON SOUNDING
315           MORN=.FALSE.
316           AFT=.TRUE.
317           CALL FINDU(JD,ISR,ISS,IDU(J),IUPNT(J),IUENDT(J),MORN,AFT,
318          +           IHRAX,HM,NOBAX,THTAPX,THTAX,TMPAX,PTAX)
319           IF(.NOT.AFT) GO TO 500
320           CALL CUP(MORN,AFT,NOBAX,THTAX,TMPAX,COEFAX)
321     C              *FIND MORNING SOUNDING FOR NEXT DAY.
322           IF(JD.LT.IJE) THEN
323             JULM=JD+1
324             MORN=.TRUE.
325             AFT=.FALSE.
326             CALL FINDU(JULM,ISR,ISS,IDU(J),IUPNT(J),IUENDT(J),MORN,AFT,
327          +             IHRM1X,HM,NOBM1X,BL,THTM1X,TMPM1X,BL)
328             IF(.NOT.MORN) GO TO 500
329             CALL CUP(MORN,AFT,NOBM1X,THTM1X,TMPM1X,COFM1X)
330           ENDIF
331     C
332     C              *WRITE TEMP SOUNDINGS TO 2-D ARRAYS
333           COEFM(J)=COEFMX
334           COEFA(J)=COEFAX
335           COEFM1(J)=COFM1X
336           ISRS(J)=ISR
337           ISSS(J)=ISS
338           IHRM(J)=IHRMX
339           IHRA(J)=IHRAX
340           NOBM(J)=NOBMX
341           NOBA(J)=NOBAX
342           NOBM1(J)=NOBM1X
343           DO 94 JJ=1,NOBMX
344             THTMM(JJ,J)=THTMMX(JJ)
345             THTM(JJ,J)=THTMX(JJ)
346             TMPM(JJ,J)=TMPMX(JJ)
347             PTM(JJ,J)=PTMX(JJ)
348          94 CONTINUE
349             DO 95 JJ=1,NOBAX
350               THTAP(JJ,J)=THTAPX(JJ)
351               THTA(JJ,J)=THTAX(JJ)
352               TMPA(JJ,J)=TMPAX(JJ)
353               PTA(JJ,J)=PTAX(JJ)
354          95 CONTINUE
355             DO 96 JJ=1,NOBM1X
356               THTM1(JJ,J)=THTM1X(JJ)
357               TMPM1(JJ,J)=TMPM1X(JJ)
358          96 CONTINUE
359     C
360             NACT=NACT+1
361             IS(NACT)=J
362             GO TO 600
363     C
364     C              ***END OF LOOP ON NEAREST STATION INDEX
365         500 CONTINUE
366     C              **NO SOUNDINGS FOUND TO APPLY TO VALLEY.
367           WRITE(6,941) NV,JD
368           STOP 'COEFF'
369     C
370         600 CONTINUE
371     C
372     C              ***COMPUTE COUPLING COEFFICIENT FOR EACH VALLEY
```

```
373     C               ***SEGMENT FROM SUNRISE ON DAY 'JD' TO SUNRISE ON
374     C               ***DAY 'JD+1'.
375     C
376     C
377     C               **COMPUTE SUNRISE AND SUNSET AT VALLEY CENTER AND
378     C               **CONVERT TO BASE TIME ZONE.
379           TSN=30.
380           CALL SOLAR(JD,TSN,XLATV(NV),YLONGV(NV),ITZV(NV),A1,ALPHA,TSR,TSS)
381           ISRV=NINT(TSR)+(ITZV(NV)-IBTZ)
382           ISSV=ISRV+NINT(TSS-TSR)
383     C
384     C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
385           IF(DETAIL) THEN
386             WRITE(LUO,61) NAMVAL(NV),ISRV,ISSV
387             WRITE(LUO,62) NAME(J),ISRS(J),ISSS(J),AOU(J)
388             WRITE(LUO,63) COEFM(J),IHRM(J)
389             WRITE(LUO,64) (THTMM(JJ,J),JJ=1,NOBM(J))
390             WRITE(LUO,65) (THTM(JJ,J),JJ=1,NOBM(J))
391             WRITE(LUO,66) (TMPM(JJ,J),JJ=1,NOBM(J))
392             WRITE(LUO,67) (PTM(JJ,J),JJ=1,NOBM(J))
393             WRITE(LUO,68) COEFA(J),IHRA(J)
394             WRITE(LUO,64) (THTAP(JJ,J),JJ=1,NOBA(J))
395             WRITE(LUO,65) (THTA(JJ,J),JJ=1,NOBA(J))
396             WRITE(LUO,66) (TMPA(JJ,J),JJ=1,NOBA(J))
397             WRITE(LUO,67) (PTA(JJ,J),JJ=1,NOBA(J))
398             WRITE(LUO,69) COEFM1(J)
399             WRITE(LUO,65) (THTM1(JJ,J),JJ=1,NOBM1(J))
400             WRITE(LUO,66) (TMPM1(JJ,J),JJ=1,NOBM1(J))
401           ENDIF
402     C*******************************************************************************
403     C
404     C               **FILL DAYTIME COEF ARRAY FROM SR+1 TO SS.
405           IF(COEFM(J).EQ.0.) THEN
406     C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
407           IF(DETAIL) THEN
408             WRITE(LUO,70)
409           ENDIF
410     C*******************************************************************************
411           TAU=(TSS-TSR)*3600.
412     C               *DETERMINE WARMING ABOVE VALLEY 'BTA'
413           DO 34 JJ=1,NOBM(J)
414             THTMX(JJ)=THTM(JJ,J)
415             THTMMX(JJ)=THTMM(JJ,J)
416             TMPMX(JJ)=TMPM(JJ,J)
417             PTMX(JJ)=PTM(JJ,J)
418        34   CONTINUE
419           DO 35 JJ=1,NOBA(J)
420             THTAX(JJ)=THTA(JJ,J)
421             PTAX(JJ)=PTA(JJ,J)
422        35   CONTINUE
423           CALL BETA(NOBM(J),NOBA(J),THTMX,THTAX,PTMX,PTAX,IHRM(J),
424          +          IHRA(J),ELVAL(NV),DPVAL(NV),BTA)
425           DO 980 NS=1,NSEG(NV)
426     C               *COMPUTE AVERAGE PRESSURE, DENSITY, AND DTHETA/DZ
427     C               *THROUGH VALLEY ATMOSPHERE USING MORNING SOUNDING
428     C               *FROM NEAREST STATION.
429           CALL VALGAM(NOBM(J),THTMX,THTMMX,TMPMX,PTMX,TOSEG(NS,NV),
430          +            ELSEG(NS,NV),PRESS,RHO,VGAMAX(NV),GAMMA)
431     C               *COMPUTE FRACTION OF POLLUTANT MASS ESCAPING VALLEY
432     C               *DURING EACH TIME STEP AFTER SUNRISE.  THE TIME STEP IS
433     C               *ASSUMED TO BE 1 HR.  THE CODE NEEDS CHECKED TO
434     C               *IMPLEMENT SHORTER TIME STEPS.
435           ALPHA1=AL1SEG(NS,NV)
```

```
435             ALPHA2=AL2SEG(NS,NV)
436             HZERO=TOSEG(NS,NV)-ELSEG(NS,NV)
437             W=WSEG(NS,NV)
438             CALL DESCNT(AOU(J),A1,ALPHA1,ALPHA2,BTA,GAMMA,HZERO,W,PRESS,
439          +                  RHO,TAU,DTMO,NMST,FRAC)
440     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
441           IF(DETAIL) THEN
442             WRITE(LUO,71) NS,AOU(J),A1,BTA,GAMMA,PRESS,RHO,TAU,NMST
443           ENDIF
444     C*******************************************************************************
445             NEND=ISRV+NMST
446             IF(NEND.GT.ISSV) THEN
447               WRITE(6,919) JD,NV
448               STOP 'COEFF'
449             ENDIF
450             JJ=0
451             DO 920 NH=ISRV+1,NEND
452               JJ=JJ+1
453               COEF2(NH,NS,NV)=FRAC(JJ)
454      920    CONTINUE
455             IF(NEND.LT.ISSV) THEN
456               DO 921 NH=NEND+1,ISSV
457                 COEF2(NH,NS,NV)=1.
458      921        CONTINUE
459             ENDIF
460      980    CONTINUE
461           ENDIF
462     C
463           IF(COEFM(J).EQ.1.) THEN
464             DO 970 NS=1,NSEG(NV)
465             DO 960 NH=ISRV+1,ISSV
466               COEF2(NH,NS,NV)=1.
467      960    CONTINUE
468      970    CONTINUE
469           ENDIF
470     C
471     C                 *FILL NIGHTIME PERIOD FROM SS+1 TO SR OF NEXT DAY.
472           IF(JD.EQ.IJE) GO TO 999
473           JULM=JD+1
474           TSN=30.
475           CALL SOLAR(JULM,TSN,XLATV(NV),YLONGV(NV),ITZV(NV),A1,ALPHA,
476          +           TSR,TSS)
477           ISRVND=NINT(TSR)+(ITZV(NV)-IBTZ)
478           IF(COEFA(J).EQ.0.) THEN
479             IF(COEFM1(J).EQ.0.) THEN
480               DO 971 NS=1,NSEG(NV)
481               DO 961 NH=ISSV+1,ISRVND+24
482                 COEF2(NH,NS,NV)=0.
483      961        CONTINUE
484      971        CONTINUE
485             ELSE
486               NT1=((ISRVND+24)-ISSV)/2+ISSV
487               DO 972 NS=1,NSEG(NV)
488               DO 962 NH=ISSV+1,NT1-2
489                 COEF2(NH,NS,NV)=0.
490      962        CONTINUE
491      972        CONTINUE
492               DO 977 NS=1,NSEG(NV)
493               DO 967 NH=NT1-1,NT1+1
494                 COEF2(NH,NS,NV)=1./3.
495      967        CONTINUE
496      977        CONTINUE
```

```
497                 DO 973 NS=1,NSEG(NV)
498                 DO 963 NH=NT1+2,ISRVND+24
499                   COEF2(NH,NS,NV)=1.
500       963       CONTINUE
501       973       CONTINUE
502               ENDIF
503             ENDIF
504             IF(COEFA(J).EQ.1.) THEN
505               IF(COEFM1(J).EQ.1.) THEN
506                 DO 974 NS=1,NSEG(NV)
507                 DO 964 NH=ISSV+1,ISRVND+24
508                   COEF2(NH,NS,NV)=1.
509       964       CONTINUE
510       974       CONTINUE
511               ELSE
512                 NT1=ISSV+2
513                 DO 975 NS=1,NSEG(NV)
514                 DO 965 NH=ISSV+1,NT1
515                   COEF2(NH,NS,NV)=1.
516       965       CONTINUE
517       975       CONTINUE
518                 DO 976 NS=1,NSEG(NV)
519                 DO 966 NH=NT1+1,ISRVND+24
520                   COEF2(NH,NS,NV)=0.
521       966       CONTINUE
522       976       CONTINUE
523               ENDIF
524             ENDIF
525     C
526     C             ***END OF VALLEY LOOP
527       999 CONTINUE
528     C
529     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
530           IF(DETAIL) THEN
531             WRITE(LUO,72)
532           ENDIF
533     C*******************************************************************************
534     C
535     C             ***WRITE OUT DECUP COEFF FOR EACH SEGMENT FOR EACH VALLEY
536           ISTART=1
537           ISTOP=24
538           IF(JD.EQ.IJS) ISTART=IHS
539           IF(JD.EQ.IJE) ISTOP=IHE
540           DO 1010 NH=ISTART,ISTOP
541             ICLK=ICLKK+NH
542             IHR1=NH+24
543             IHR2=NH
544             DO 1005 NV=1,NVALE
545               IF(COEF2(IHR2,1,NV).EQ.-999.) THEN
546                 WRITE(27) ICLK,IDVAL(NV),(COEF1(IHR1,NS,NV),NS=1,NSEG(NV))
547               ELSE
548                 WRITE(27) ICLK,IDVAL(NV),(COEF2(IHR2,NS,NV),NS=1,NSEG(NV))
549               ENDIF
550       1005    CONTINUE
551       1010 CONTINUE
552     C
553     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
554           IF(DETAIL) THEN
555             DO 1015 NV=1,NVALE
556               WRITE(LUO,73) NAMVAL(NV)
557               DO 1020 NH=ISTART,ISTOP
558                 IHR1=NH+24
```

```
559                 IHR2=NH
560                 IF(COEF2(IHR2,1,NV).EQ.-999.) THEN
561                   WRITE(LUO,74) NH,(COEF1(IHR1,NS,NV),NS=1,NSEG(NV))
562                 ELSE
563                   WRITE(LUO,74) NH,(COEF2(IHR2,NS,NV),NS=1,NSEG(NV))
564                 ENDIF
565       1020    CONTINUE
566       1015  CONTINUE
567           ENDIF
568     C*************************************************************************
569     C
570     C           ***REDEFINE COEF1 ARRAY TO COEF2 ARRAY.
571           DO 41 NV=1,NVALE
572           DO 41 NS=1,NSEG(NV)
573           DO 41 NH=1,34
574             COEF1(NH,NS,NV)=COEF2(NH,NS,NV)
575             COEF2(NH,NS,NV)=-999.
576        41 CONTINUE
577     C
578     C           ***END OF JULIAN DAY LOOP
579       900 CONTINUE
580     C
581     C
582           RETURN
583         5 FORMAT(1H1,///,5X,'TRACE OF SUBROUTINE COEFF EXECUTION',
584          +' - COEFF COMPUTES THE COUPLING COEFFICIENT FOR ALL VALLEY',/
585          +,5X,'SEGMENTS FOR EACH PROCESSING TIME STEP OF',F6.0,' SEC.',
586          +'  THE BASE TIME ZONE IS ',A10,'.')
587        10 FORMAT(1H0,/,5X,'THE VALLEYS PROCESSED ARE:',/,10X,'NAME',
588          +6X,'ID NO',2X,'X (KM)',2X,'Y (KM)',2X,'LAT (DEG)',2X,
589          +'LONG (DEG)',2X,'ELEV (M MSL)',2X,'DEPTH (M)',2X,'WS (M/S)',
590          +2X,'GAM (DEGK/M)',2X,'# SEGS')
591        12 FORMAT(1H ,6X,A10,2X,I6,3X,F5.1,3X,F5.1,3X,F7.3,5X,F7.3,6X,F6.1,
592          +7X,F5.0,6X,F4.1,8X,F4.3,8X,I2)
593        14 FORMAT(1H0,/,5X,'THE SEGMENT DATA FOR VALLEY "',A10,
594          +'" ARE:',/,8X,'ID SEG',2X,'X (KM)',2X,'Y (KM)',2X,'EL (M MSL)'
595          +,2X,'TOP EL (M MSL)',2X,'WIDTH (M)',2X,'ANG1 (DEG)',2X,
596          +'ANG2 (DEG)')
597        15 FORMAT(1H ,9X,I3,4X,F5.1,3X,F5.1,4X,F6.1,8X,F6.1,8X,F5.0,8X,F3.0,
598          +9X,F3.0)
599       360 FORMAT(I6,1X,A10,I3,2F10.3,I3,2F8.4,2F6.0,I3)
600       361 FORMAT(2I6,2F10.3,I3,5F6.0)
601        53 FORMAT(1H0,2X,'VALLEY SEGMENTS INFO BEING READ IS NOT',
602          +                ' FOR VALLEY ID',I4,' DESIRED.')
603       941 FORMAT(1H0,' NO SOUNDINGS FOR VALLEY',I4,' ON DAY',I4,' .')
604       919 FORMAT(1H0,' INVERSION BREAKUP FOR VALLEY',I4,' WENT PAST',
605          +            ' SUNSET ON DAY',I4,' .')
606        60 FORMAT(1H0,/,5X,'JULIAN DAY =',I4,10X,'(OUTPUT FROM SUBROUTINE',
607          +' COEFF)')
608        61 FORMAT(1H ,/,7X,'VALLEY = ',A10,14X,'VALLEY SUNRISE =',I3,
609          +6X,'VALLEY SUNSET =',I3)
610        62 FORMAT(1H ,6X,'NEAREST STATION = ',A10,5X,'STATION SUNRISE',
611          +' =',I3,5X,'STATION SUNSET =',I3,5X,'A0 = ',F5.3)
612        63 FORMAT(1H ,/,9X,'MORNING SOUNDING',7X,'COUPLING COEFF =',F3.0,5X,
613          +'HOUR OF SOUNDING =',I3)
614        64 FORMAT(1H ,9X,'PRESS (MB) =',6X,15F7.0)
615        65 FORMAT(1H ,9X,'HEIGHT (MMSL) =',3X,15F7.0)
616        66 FORMAT(1H ,9X,'TEMP (DEGC) =',5X,15F7.1)
617        67 FORMAT(1H ,9X,'POT TEMP (DEGK) =',1X,15F7.1)
618        68 FORMAT(1H ,/,9X,'AFTERNOON SOUNDING',5X,'COUPLING COEFF =',F3.0,
619          +5X,'HOUR OF SOUNDING =',I3)
620        69 FORMAT(1H ,/,9X,'MORN SND (DAY+1)',7X,'COUPLING COEFF =',F3.0)
```

```
621         70 FORMAT(1H0,8X,'MORN SOUNDING IS DECOUPLED, THEREFORE "DESCNT" IS',
622           +' CALLED WITH THE FOLLOWING INPUTS AND RESULT:',/,15X,'SEG',11X,
623           +'A0',7X,'A1(W/M**2)',2X,'BETA(DEGK/S)',1X,'GAMA(DEGK/M)',2X,
624           +'PRESS(MB)',3X,'RHO(KG/M**3)',4X,'TAU(S)',4X,'# TIME STEPS')
625         71 FORMAT(1H ,13X,I3,9X,F5.3,8X,F5.0,7X,F7.5,7X,F6.4,7X,F5.0,8X,
626           +F5.3,8X,F6.0,9X,I2)
627         72 FORMAT(1H0,6X,'THE HOURLY COUPLING COEFF FOR ALL SEGMENTS OF ALL',
628           +' VALLEYS ON THIS JULIAN DAY ARE:')
629         73 FORMAT(1H0,8X,'THE VALLEY IS',2X,A10,//,12X,'HOUR ',17X,
630           +'VALLEY SEGMENTS')
631         74 FORMAT(1H ,11X,I3,5X,10F8.4)
632            END
```

```
001     C  SUBROUTINE: MET.CUP
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO COMPUTE THE 'MORNING' OR 'AFTERNOON'
005     C  COUPLING COEFFICIENT FROM EITHER A 'MORNING' OR 'AFTERNOON' TEMP
006     C  SOUNDING, RESPECTIVELY.  A COUPLING COEFFICIENT OF '1' MEANS A
007     C  VALLEY ATMOSPHERE IS COUPLED WITH THE FLOW ABOVE RIDGETOPS, AND A
008     C  COUPLING COEFFICIENT OF '0' MEANS THE VALLEY ATMOSPHERE IS
009     C  DECOUPLED.
010     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
011     C  THE INPUTS ARE:
012     C       MORN = LOGICAL VARIABLE IF .TRUE. MEANS FIND 'MORNING' COEF
013     C       AFT = LOGICAL VARIABLE IF .TRUE. MEANS FIND 'AFTERNOON' COEF
014     C       NOBU = NUMBER OF OBSERVATIONS IN SOUNDING
015     C       THTM = ARRAY OF OBS. HEIGHTS FOR SND. (M MSL)
016     C       TMP = ARRAY OF TEMP. OBS. FOR SND. (DEG C)
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       COEF = COUPLING COEFFICIENT
020     C            = 1  [MEANS VALLEY ATMOS COUPLED WITH FLOW ABOVE RIDGES]
021     C            = 0  [MEANS VALLEY ATMOS DECOUP WITH FLOW ABOVE RIDGES]
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLING SUBROUTINES:
024     C       COEFF
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  CALLED SUBROUTINES:
027     C       TPROF
028     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
029     C  COMMON BLOCKS:
030     C       (NONE)
031     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
032     C  PARAMETERS DEFINITIONS:
033     C       NOB = NUMBER OF OBSERVATIONS IN UPPER AIR SOUNDINGS
034     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
035     C
036             SUBROUTINE CUP(MORN,AFT,NOBU,THTM,TMP,COEF)
037     C
038             PARAMETER
039            +    NOB=30
040     C
041             LOGICAL
042            +    MORN,           AFT
043     C
044             DIMENSION
045            +    THTM(NOB),      TMP(NOB)
046     C
047     C
048             IF(MORN) THEN
049     C                ***COMPUTE CUP COEF FROM MORN TEMP SOUNDING
050               HT=THTM(1)+500.
051               CALL TPROF(NOBU,THTM,TMP,HT,T500)
052     C                *DETERMINE MAX TEMP. IN LOWEST 500 M.
053               TMAX=T500
054               DO 2 J=1,NOBU
055                 IF(THTM(J).GT.HT) GO TO 3
056                 IF(TMP(J).GT.TMAX) TMAX=TMP(J)
057           2   CONTINUE
058           3   CONTINUE
059               IF(TMAX.GT.TMP(1)+1.) THEN
060                 COEF=0.
061               ELSE
062                 COEF=1.
```

```
063             ENDIF
064           ENDIF
065     C
066           IF(AFT) THEN
067     C                *COMPUTE CUP COEF FROM AFT TEMP SOUNDING
068             HTL=THTM(1)+75.
069             HTU=THTM(1)+500.
070             CALL TPROF(NOBU,THTM,TMP,HTL,T75)
071             CALL TPROF(NOBU,THTM,TMP,HTU,T500)
072             IF(T75-T500.GE.2.1) THEN
073               COEF=1.
074             ELSE
075               COEF=0.
076             ENDIF
077           ENDIF
078     C
079           RETURN
080           END
```

```
001     C  SUBROUTINE: MET.DESCNT
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO COMPUTE THE FRACTION OF VALLEY
005     C  CROSS-SECTIONAL AREA CONSUMED BY CBL GROWTH AND INV. TOP
006     C  DESCENT FOR EACH MET MODEL TIME STEP DURING THE MORNING
007     C  TRANSITION PERIOD.  THE SUM OF THE FRACTIONS FOR ALL MET
008     C  MODEL TIME STEPS DURING INVERSION DESTRUCTION SHOULD TOTAL 1.0 .
009     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
010     C  THE INPUTS ARE:
011     C       A0 = FRACTION (0-1) OF A1 CONVERTED TO SENSIBLE HEAT FLUX
012     C       A1 = SOLAR FLUX ON HORIZ SFC AT SOLAR NOON (W/M2)
013     C       ALPHA1 = SIDEWALL NO. 1 INCLINATION ANGLE (DEG)
014     C       ALPHA2 = SIDEWALL NO. 2 INCLINATION ANGLE (DEG)
015     C       BETA = RATE OF TEMP CHANGE AT INVERSION TOP (K/SEC)
016     C       DTMO = MET. MODEL TIME STEP (600 TO 3600 SEC)
017     C       GAMMA = VERT. POT. TEMPERATURE GRADIENT IN INVERSION (K/M)
018     C       HZERO = INITIAL HEIGHT OF INVERSION TOP (M)
019     C       PRESS = AVG. PRESSURE (MB) AT INV. CENTER AT SUNRISE
020     C       RHO = AIR DENSITY (KG/M3)
021     C       TAU = LENGTH OF DAYLIGHT PERIOD (SEC). 43200SEC=12HR
022     C       W = VALLEY FLOOR WIDTH (M)
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  THE OUTPUTS ARE:
025     C       FRAC = FRACTIONAL AMOUNT OF VALLEY AREA CROSS-SECTION CONSUMED
026     C              BY CBL GROWTH AND INVERSION TOP DESCENT DURING ONE MET
027     C              MODEL TIME STEP.  FRAC IS AN ARRAY OF FRACTIONS, EACH
028     C              ELEMENT FOR A MET MODEL TIME STEP UNTIL INVERSION IS
029     C              DESTROYED.  THE SUM OF THE FRACTIONS FOR ALL MET TIME
030     C              STEP TO INV. DESTRUCTION SHOULD TOTAL 1.0 .
031     C       NMST = NUMBER OF MET MODEL TIME STEPS FOR INV. DESTRUCTION
032     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
033     C  CONSTANT DEFINITIONS:
034     C       CP = 1005.  [SPECIFIC HEAT AT CONSTANT PRESSURE (J/KG/K)]
035     C       DELTAT = 600.  [INVERSION DESTRUCTION TIME STEP (SEC)]
036     C       HCBLI = 25.  [INITIAL CBL HEIGHT (M)]
037     C       PI = 3.14159  [TRIGONOMETRIC CONSTANT]
038     C       RK = 0.15  FRCTN. (0-1) OF SENS. HT. FLUX GOING INTO CBL GROWTH
039     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
040     C  CALLING SUBROUTINES:
041     C       COEFF
042     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
043     C  CALLED SUBROUTINES:
044     C       (NONE)
045     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
046     C  COMMON BLOCKS:
047     C       (NONE)
048     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
049     C  PARAMETER DEFINITIONS:
050     C       NS = NUMBER OF MET MODEL TIME STEPS FOR INVERSION DESTRUCTION
051     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
052     C
053           SUBROUTINE DESCNT(A0,A1,ALPHA1,ALPHA2,BETA,GAMMA,HZERO,W,PRESS,
054          +                  RHO,TAU,DTMO,NMST,FRAC)
055     C
056           PARAMETER NS=60
057     C
058           DIMENSION FRAC(NS)
059     C
060     C**** INPUT PARAMETERS
061           CP=1005.
062           PI=3.14159
```

```
063           DELTAT=600.
064           RK=0.15
065           HCBLI=25.
066           N=0
067           TI=0.
068           T=0.
069           DTOR=PI/180.
070           AL1=ALPHA1*DTOR
071           AL2=ALPHA2*DTOR
072           II=0
073           IF(A0.LT..0001) A0=.0001
074           IF(GAMMA.LT..0001) GAMMA=.0001
075     C
076     C**** CALCULATE SECONDARY PARAMETERS
077           FACT1=A0*A1/(RHO*CP)
078           THTOT=(1000./PRESS)**.286
079           C=(1./TAN(AL1))+(1./TAN(AL2))
080           NM=NINT(DTMO/DELTAT)
081           IIMAX=NINT(TAU/DTMO)
082     C
083     C**** DO CALCULATIONS
084           HTOP=HZERO
085           HCBL=HCBLI
086           DELT=HTOP-HCBL
087           VNOT=DELT*W+C*DELT/2.
088           VOLP=VNOT
089           FRCTOT=0.
090         1 CONTINUE
091           T=T+DELTAT
092           N=N+1
093           ITEST=MOD(N,NM)
094           FACT2=FACT1*SIN(PI*(T-TI)/TAU)
095     C
096           IF(RK.NE.0.) THEN
097              DHDT=FACT2*THTOT*RK*(W+HCBL*C)/(GAMMA*HCBL*(W+(HCBL*C)/2.))
098           ELSE
099              DHDT=0.
100           ENDIF
101     C
102           FNUM=(W+HTOP*C-RK*(W+HCBL*C))*FACT2-((BETA*
103          +(HZERO-HTOP)*(W+((HZERO+HTOP)/2.)*C))/(2.*THTOT))
104           FDENOM=HTOP*GAMMA*(W+(HTOP*C)/2.)+(BETA*(T-TI)*(W+HTOP*C))/2.
105           DHTDT=-THTOT*FNUM/FDENOM
106           DHT=DHTDT*DELTAT
107           HTOP=HTOP+DHT
108           DHC=DHDT*DELTAT
109           HCBL=HCBL+DHC
110           IF(ITEST.EQ.0) THEN
111             II=II+1
112             DELT=HTOP-HCBL
113             IF(DELT.GT.0.) THEN
114               VOL=DELT*W+C*DELT/2.
115             ELSE
116               VOL=0.
117             ENDIF
118             FRAC(II)=(VOLP-VOL)/VNOT
119             VOLP=VOL
120             FRCTOT=FRCTOT+FRAC(II)
121             IF(II.EQ.IIMAX) THEN
122               DO 10 J=1,IIMAX
123               FRAC(J)=FRAC(J)/FRCTOT
124      10       CONTINUE
```

```
125               NMST=IIMAX
126               RETURN
127             ENDIF
128           ENDIF
129     C
130     C**** STOP CALCULATIONS WHEN INVERSION TOP SINKS BELOW CBL TOP
131           IF(HTOP.LE.HCBL) THEN
132             IF(II.EQ.0) THEN
133               II=1
134               FRAC(II)=.9999
135               NMST=II
136               RETURN
137             ENDIF
138             IF(ITEST.NE.0) THEN
139               FR=VOLP/VNOT
140               FRCTOT=FRCTOT+FR
141               IF(FLOAT(ITEST).LT.FLOAT(NM)/2.) THEN
142                 FRAC(II)=FRAC(II)+FR
143               ELSE
144                 II=II+1
145                 FRAC(II)=FR
146               ENDIF
147             ENDIF
148             NMST=II
149             RETURN
150           ELSE
151              GO TO 1
152           ENDIF
153     C
154           END
```

```
001     C  SUBROUTINE: MET.DEW
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO COMPUTE THE DEW POINT TEMPERATURE
005     C  AT A POINT GIVEN THE RELATIVE HUMIDITY AND THE TEMPERATURE.
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C       RH = RELATIVE HUMIDITY (%)
009     C       T = TEMPERATURE (DEG C)
010     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
011     C  THE OUTPUTS ARE:
012     C       D = DEW POINT TEMPERATURE (DEG C)
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  CALLING SUBROUTINES:
015     C       UPPER
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  CALLED SUBROUTINES:
018     C       (NONE)
019     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
020     C  COMMON BLOCKS:
021     C       (NONE)
022     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
023     C
024           SUBROUTINE DEW (T,RH,DP)
025     C
026     C          ***COMPUTE THE SATURATION VAPOR PRESSURE (ES).
027           ES=6.1078*10.**(7.5*T/(237.3+T))
028     C          ***COMPUTE THE ACTUAL VAPOR PRESSURE (E).
029           E=ES*RH/100.
030     C          ***COMPUTE THE DEW POINT TEMPERATURE.
031           A=ALOG10(E/6.1078)
032           DP=(237.3*A)/(7.5-A)
033     C
034           RETURN
035           END
```

```
001     C  SUBROUTINE: MET.FILLMB
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE PRESSURE FOR AN
005     C  OBSERVATION POINT '2' IN A SOUNDING GIVEN THE TEMPERATURE, DEW
006     C  POINT, AND HEIGHT AT THIS POINT PLUS THE PRESSURE, TEMPERATURE,
007     C  DEW POINT, AND HEIGHT AT A LOWER POINT '1'.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       D1 = DEW POINT TEMP. AT OBS. BELOW MISSING PRESS. (DEG C)
011     C       D2 = DEW POINT TEMP. AT OBS. OF MISSING PRESSURE (DEG C)
012     C       P1 = PRESSURE AT OBS. BELOW MISSING PRESS. (MB)
013     C       T1 = TEMPERATURE AT OBS. BELOW MISSING PRESS. (DEG C)
014     C       T2 = TEMPERATURE AT OBS. OF MISSING PRESSURE (DEG C)
015     C       Z1 = HEIGHT AT OBS. BELOW MISSING PRESS. (M MSL)
016     C       Z2 = HEIGHT AT OBS OF MISSING PRESSURE (M MSL)
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       P2 = PRESSURE AT OBS. OF MISSING PRESSURE (MB)
020     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
021     C  CONSTANT DEFINITIONS:
022     C       C = 0.06815  [2*(ACCEL. DUE TO GRAV)/(UNIV. GAS CONST.)]
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLING SUBROUTINES:
025     C       UPPER
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLED SUBROUTINES:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  COMMON BLOCKS:
031     C       (NONE)
032     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
033     C
034           SUBROUTINE FILLMB (Z2,Z1,P2,P1,T2,T1,D2,D1)
035     C
036     C           ***CONSTANTS
037           C=0.06815
038     C
039     C           ***COMPUTE SECONDARY VARIABLES: ACTUAL VAPOR PRESSURE
040     C           ***(E1) AND, VIRTUAL TEMPERATURE (TSTAR1) AT LOWER LEVEL.
041           E1=6.1078*10.**(7.5*D1/(237.3+D1))
042           TSTAR1=(T1+273.15)*((P1+.000798*E1)/(P1-.378*E1))
043     C           ***COMPUTE ACTUAL VAPOR PRESSURE AT UPPER LEVEL (2).
044           E2=6.1078*10.**(7.5*D2/(237.3+D2))
045           DELTAZ=Z2-Z1
046     C
047     C           ***CONVERGE TO A SOLUTION USING NEWTON'S METHOD.
048           P2=P1
049           TK2=T2+273.15
050           F1E2=.000798*E2
051           F2E2=.378*E2
052           IT=0
053         5 IT=IT+1
054           TSTAR2=TK2*((P2+F1E2)/(P2-F2E2))
055           TSTR2P=-TK2*((F1E2+F2E2)/(P2-F2E2))
056           A=C*DELTAZ/(TSTAR2+TSTAR1)
057           AP=-A*TSTR2P/(TSTAR2+TSTAR1)
058           ETA=EXP(-A)
059           G=P2-P1*ETA
060           GP=1.+P1*AP*ETA
061           FRAC=G/GP
062           P2=P2-FRAC
```

```
063           CK=ABS(FRAC)
064           IF(CK.LE..05) RETURN
065           IF(IT.GT.10)THEN
066             WRITE(6,6)
067             STOP 'FILLMB'
068           ENDIF
069           GO TO 5
070     C
071         6 FORMAT(1H ,5X,'MISSING PRESSURE NOT FOUND IN 10 ITERATIONS.')
072           END
```

```
001     C  SUBROUTINE: MET.FILLME
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE HEIGHT FOR AN
005     C  OBSERVATION POINT '2' IN A SOUNDING GIVEN THE TEMPERATURE, DEW
006     C  POINT, AND PRESSURE AT THIS POINT PLUS THE PRESSURE, TEMPERATURE,
007     C  DEW POINT, AND HEIGHT AT A LOWER POINT '1'.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       D1 = DEW POINT TEMP. AT OBS. BELOW MISSING HEIGHT (DEG C)
011     C       D2 = DEW POINT TEMP. AT OBS. OF MISSING HEIGHT (DEG C)
012     C       P1 = PRESSURE AT OBS. BELOW MISSING HEIGHT (MB)
013     C       P2 = PRESSURE AT OBS. OF MISSING HEIGHT (MB)
014     C       T1 = TEMPERATURE AT OBS. BELOW MISSING HEIGHT (DEG C)
015     C       T2 = TEMPERATURE AT OBS. OF MISSING HEIGHT (DEG C)
016     C       Z1 = HEIGHT AT OBS. BELOW MISSING HEIGHT (M MSL)
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       Z2 = HEIGHT AT OBS WITH MISSING HEIGHT (M MSL)
020     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
021     C  CONSTANT DEFINITIONS:
022     C       C = 0.06815  [2*(ACCEL. DUE TO GRAV)/(UNIV. GAS CONST.)]
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLING SUBROUTINES:
025     C       UPPER
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLED SUBROUTINES:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  COMMON BLOCKS:
031     C       (NONE)
032     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
033     C
034           SUBROUTINE FILLME (Z2,Z1,P2,P1,T2,T1,D2,D1)
035     C
036     C           ***CONSTANTS
037           C=0.06815
038     C
039     C           ***COMPUTE SECONDARY VARIABLES: ACTUAL VAPOR PRESSURE
040     C           ***(E) AND, VIRTUAL TEMPERATURE (TSTAR).
041           E1=6.1078*10.**(7.5*D1/(237.3+D1))
042           E2=6.1078*10.**(7.5*D2/(237.3+D2))
043           TSTAR1=(T1+273.15)*((P1+.000798*E1)/(P1-.378*E1))
044           TSTAR2=(T2+273.15)*((P2+.000798*E2)/(P2-.378*E2))
045     C
046           Z2=Z1+((TSTAR2+TSTAR1)/C)*ALOG(P1/P2)
047     C
048           RETURN
049           END
```

```
001     C  SUBROUTINE: MET.FINDU
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO READ IN A 'MORNING' OR 'AFTERNOON'
005     C  TEMPERATURE SOUNDING FROM THE DIRECT ACCESS DISK FILE ON UNIT 9.
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C    THRU ARGUMENT:
009     C      JD = JULIAN DATE THAT SOUNDING IS DESIRED ON
010     C      ISR = TIME OF SUNRISE ON 'JD' (1 TO 24 LST)
011     C      ISS = TIME OF SUNSET ON 'JD' (1 TO 24 LST)
012     C      IDU = ID OF DESIRED UPPER AIR STATION
013     C      IUPNT = POINTER INDICATING RECORD NUMBER OF TEMP SNDG ON
014     C              DIRECT ACCESS DISK FILE ON UNIT 9 TO START READING FROM
015     C              FOR DESIRED STATION.
016     C      IUENDT = ENDING RECORD NUMBER OF TEMP SNDGS FOR DESIRED
017     C               UPPER AIR STN ON DIRECT ACCESS DISK FILE ON UNIT 9
018     C      MORN = LOGICAL VARIABLE IF .TRUE. MEANS FIND 'MORNING' SOUNDING
019     C      AFT = LOGICAL VARIABLE IF .TRUE. MEANS FIND 'AFTERNOON' SND
020     C    THRU LOGICAL UNIT 9:
021     C      IDD = ID NUMBER OF STATION AT SOUNDING
022     C      IJUL = JULIAN DATE OF SOUNDING
023     C      IHR = HOUR-OF-DAY OF SOUNDING
024     C      HM = HEIGHT OF MECHANICAL MIXING DEPTH (M)
025     C      NOBU = NUMBER OF OBSERVATIONS IN SOUNDING
026     C      THTP = ARRAY OF OBS. PRES. FOR SND. (MB)
027     C      THTM = ARRAY OF OBS. HEIGHTS FOR SND. (M MSL)
028     C      TMP = ARRAY OF TEMP. OBS. FOR SND. (DEG C)
029     C      PT = ARRAY OF POT. TEMP. OBS. FOR SND (DEG K)
030     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
031     C  THE OUTPUTS ARE:
032     C    THRU ARGUMENT:
033     C      IUPNT = POINTER INDICATING LAST RECORD NUMBER OF TEMP SNDG ON
034     C              DIRECT ACCESS DISK FILE ON UNIT 9 WHICH WAS READ FROM
035     C      MORN = LOGICAL VARIABLE IF .TRUE. MEANS 'MORNING' SOUNDING WAS
036     C             FOUND
037     C      AFT = LOGICAL VARIABLE IF .TRUE. MEANS 'AFTERNOON' SND WAS FND
038     C      IHR = HOUR-OF-DAY OF SOUNDING
039     C      HM = HEIGHT OF MECHANICAL MIXING DEPTH (M)
040     C      NOBU = NUMBER OF OBSERVATIONS IN SOUNDING
041     C      THTP = ARRAY OF OBS. PRES. FOR SND. (MB)
042     C      THTM = ARRAY OF OBS. HEIGHTS FOR SND. (M MSL)
043     C      TMP = ARRAY OF TEMP. OBS. FOR SND. (DEG C)
044     C      PT = ARRAY OF POT. TEMP. OBS. FOR SND (DEG K)
045     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
046     C  CONSTANTS DEFINITIONS:
047     C      ISRB = 3  [MAX NUMBER OF HOURS BEFORE SUNRISE DURING WHICH A
048     C                 SOUNDING WILL STILL BE USED AS A 'MORNING' SOUNDING]
049     C      ISRA = 1  [MAX NUMBER OF HOURS AFTER SUNRISE DURING WHICH A
050     C                 SOUNDING WILL STILL BE USED AS A 'MORNING' SOUNDING]
051     C      ISSB = 3  [MAX NUMBER OF HOURS BEFORE SUNSET DURING WHICH A
052     C                 SOUNDING WILL STILL BE USED AS AN 'AFTERNOON' SND]
053     C      ISSA = 1  [MAX NUMBER OF HOURS AFTER SUNSET DURING WHICH A
054     C                 SOUNDING WILL STILL BE USED AS AN 'AFTERNOON' SND]
055     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
056     C  FILES READ FROM:
057     C      LOGICAL UNIT 9 = DIRECT ACCESS UNFORMATTED DISK FILE CONTAINING
058     C                       THE UPPER AIR TEMPERATURE DATA
059     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
060     C  CALLING SUBROUTINES:
061     C      COEFF, TEMP
062     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
063     C  CALLED SUBROUTINES:
064     C       (NONE)
065     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
066     C  COMMON BLOCKS:
067     C       (NONE)
068     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
069     C  PARAMETERS DEFINITIONS:
070     C       NOB = NUMBER OF OBSERVATIONS IN UPPER AIR SOUNDINGS
071     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
072     C
073           SUBROUTINE FINDU(JD,ISR,ISS,IDU,IUPNT,IUENDT,MORN,AFT,IHR,HM,
074          +                 NOBU,THTP,THTM,TMP,PT)
075     C
076           PARAMETER
077          +    NOB=30
078     C
079           LOGICAL
080          +    MORN,          AFT
081     C
082           DIMENSION
083          +    PT(NOB),       THTM(NOB),     THTP(NOB),     TMP(NOB)
084     C
085     C                ***CONSTANTS
086           ISRB=3
087           ISRA=1
088           ISSB=3
089           ISSA=1
090     C
091     C
092     C                ***SPECIFY TIME BOUNDARIES ON DEFINING WHETHER 'MORNING'
093           IF(MORN) THEN                ***OR 'AFTERNOON' SOUNDING.
094             ILOW=ISR-ISRB
095             IHIGH=ISR+ISRA
096           ENDIF
097           IF(AFT) THEN
098             ILOW=ISS-ISSB
099             IHIGH=ISS+ISSA
100           ENDIF
101     C
102     C                ***FIND SOUNDING.
103        44 IUPNT=IUPNT+1
104           IF (IUPNT.LE.IUENDT) THEN
105             READ (9'IUPNT) IDD,IJUL,IHR,HM,NOBU,(THTP(K),THTM(K),TMP(K),
106            +                PT(K),K=1,NOBU)
107             IF (IDD.NE.IDU)  THEN
108               WRITE (6,32)
109               STOP 'FINDU'
110             ENDIF
111             IF (IJUL.LT.JD) GO TO 44
112             IF (IJUL.GT.JD) THEN
113               MORN=.FALSE.
114               AFT=.FALSE.
115             ELSE
116               IF (IHR.LT.ILOW) GO TO 44
117               IF (IHR.GT.IHIGH) THEN
118                 MORN=.FALSE.
119                 AFT=.FALSE.
120               ENDIF
121             ENDIF
122           ELSE
123             MORN=.FALSE.
124             AFT=.FALSE.
```

```
125           ENDIF
126           IUPNT=IUPNT-1
127     C
128           RETURN
129        32 FORMAT (2X,'ID OF SOUNDING DOES NOT CORRESPOND WITH ID REQUIRED.')
130           END
```

```
001     C  SUBROUTINE: MET.FROUDE
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO ADJUST THE WIND FIELD OVER THE GRAMA
005     C  DOMAIN ACCORDING TO FROUDE NUMBER CONSIDERATIONS.  WINDS ARE
006     C  COMPUTED AT GRID POINTS ON ALL 'GAMMA' SURFACES USING THE AMPLITUDE
007     C  FUNCTIONS FROM THE INITIAL INPUT WINDS.  FROM UPPER AIR AND SURFACE
008     C  TEMPERATURE DATA A 'DIVIDING STREAMLINE HEIGHT' IS COMPUTED AT EACH
009     C  GRID POINT.  THE WINDS BELOW THE DIVIDING STREAMLINE HEIGHT ARE
010     C  ADJUSTED TO BE TANGENT TO THE TERRAIN SLOPE AND THEN THESE ADJUSTED
011     C  PROFILES PLUS THE ORIGINAL WIND PROFILES ARE USED TO COMPUTE NEW
012     C  AMPLITUDE FUNCTIONS WHICH DEFINES THE WIND FIELD.  THIS SUBROUTINE
013     C  COMPUTES THE WIND FIELD AT A GIVEN HOUR.
014     C    ALL COMPUTED 'SOUNDINGS', WHETHER MODIFIED OR NOT, ARE USED IN
015     C  COMPUTING THE NEW AMPLITUDE FUNCTIONS IF THE DIVIDING STREAMLINE
016     C  HEIGHT IS NONZERO.  IF THE DIVIDING STREAMLINE HEIGHT IS ZERO THEN
017     C  THE COMPUTED SOUNDING IS NOT USED.  TO ELIMINATE ALL UNADJUSTED
018     C  'SOUNDINGS' FROM COMPUTATION OF THE AMPLITUDE FUNCTIONS, UNCOMMENT
019     C  THE 7 LINES MARKED 'C!!!'.
020     C    TO GET MORE DETAILED OUTPUT FROM SUBROUTINE FROUDE, SET 'DETLF'
021     C  EQUAL TO '.TRUE.'.
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  THE INPUTS ARE:
024     C    THRU ARGUMENT:
025     C      JUL = JULIAN DATE OF DAY BEING PROCESSED
026     C      IHR = HOUR-OF-DAY BEING PROCESSED (1 TO 24)
027     C      NFS = NUMBER OF 'STATIONS' FOR COMPUTING WIND FIELD
028     C      NGAMA = NUMBER OF 'GAMMA' SURFACES TO COMPUTE WINDS ON
029     C      HTGAMA = ARRAY OF HEIGHTS OF 'GAMMA' SURFACES (TRANSF. COORD)
030     C      NTERMS = NUMBER OF POLYNOMIAL TERMS TO FIT WINDS
031     C      IDTF = IDENTIFICATION NUMBER OF FROUDE TERRAIN FILE
032     C      RAD = MAX. DISTANCE GREATER THAN WHICH DATA WILL NOT BE
033     C            INTERPOLATED TO A GRID POINT (KM)
034     C      NSTA = NUMBER OF UPPER AIR STATIONS
035     C      NSSTA = NUMBER OF SURFACE WEATHER STATIONS
036     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
037     C                              8(PACIFIC)
038     C      FRO1ST = LOGICAL VARIABLE IF .TRUE. MEANS FIRST CALL TO THIS
039     C               SUBROUTINE AND CERTAIN STEPS MUST BE TAKEN
040     C      HA = HEIGHT OF THE ADIABATIC ATMOSPHERE (M MSL)
041     C      ROA = SEA-LEVEL AIR DENSITY (KG/M**3)
042     C      LUO = LOGICAL UNIT TO WHICH PRINTOUT IS DIRECTED
043     C    THRU LOGICAL UNIT 15:
044     C      IX0F = X-COORD ORIGIN OF FROUDE GRID (KM)
045     C      IY0F = Y-COORD ORIGIN OF FROUDE GRID (KM)
046     C      NXCF = NUMBER OF GRID CELLS IN X-DIR OF FROUDE GRID
047     C      NYCF = NUMBER OF GRID CELLS IN Y-DIR OF FROUDE GRID
048     C      ICSF = CELL SIZE IN FROUDE GRID (KM)
049     C      OH = ARRAY OF OBSTACLE HEIGHTS FOR EACH FROUDE GRID POINT (M)
050     C      THET = ARRAY OF DRAINAGE DIRECTION FOR EACH FROUDE GRID POINT
051     C             (DEG - 000 NORTH)
052     C    THRU COMMON BLOCK 'MAIN2':
053     C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
054     C            1 - 'SMALL H' EQUAL TO CONSTANT
055     C            2 - 'BIG H' EQUAL TO CONSTANT
056     C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
057     C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
058     C            (SMALL H = BIG H - TERRAIN HEIGHT)
059     C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
060     C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
061     C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
062     C           TO HH)
```

```
063     C       IX0 = X-COORD OF DOMAIN ORIGIN IN GRAMA COORDS (KM)
064     C       IY0 = Y-COORD OF DOMAIN ORIGIN IN GRAMA COORDS (KM)
065     C       SLX = DOMAIN WIDTH IN X-DIR (KM)
066     C       SLY = DOMAIN WIDTH IN Y-DIR (KM)
067     C     THRU COMMON BLOCK 'AMPL1':
068     C       UP = ARRAY OF X-COMP OF WINDS AT EACH 'GAMMA' SURFACE FOR ALL
069     C            INITIAL INPUT WIND STATIONS (M/S)
070     C       VP = SAME AS 'UP' EXCEPT Y-COMP OF WINDS
071     C       XG = ARRAY OF 'ALPHA' COORD OF INPUT WIND STATIONS
072     C       YG = ARRAY OF 'BETA' COORD OF INPUT WIND STATIONS
073     C       ZSTER = ARRAY OF INPUT STATION ELEVATIONS (M MSL)
074     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
075     C  THE OUTPUTS ARE:
076     C     THRU ARGUMENT:
077     C       NTERMA = 0, MEANS NO GRID POINTS ADJUSTED FOR FROUDE NUMBER
078     C                CONSIDERATIONS; ELSE = NTERMS
079     C       FRO1ST = LOGICAL VARIABLE SET .FALSE. MEANS THE FIRST CALL TO
080     C                FROUDE HAS BEEN MADE
081     C       IC = TOTAL NUMBER OF RESULTING MODIFIED 'SOUNDINGS' INCLUDING
082     C            THE ORIGINAL SOUNDINGS
083     C     THRU COMMON BLOCK 'AMPL1':
084     C       SAME AS DESCRIBED IN 'INPUT' EXCEPT EACH VARIABLE INCLUDES
085     C       FROUDE NUMBER MODIFICATIONS
086     C     THRU COMMON BLOCK 'FROUD':
087     C       AG = ARRAY OF 'ALPHA' COORD OF FROUDE GRID POINTS
088     C       BG = ARRAY OF 'BETA' COORD OF FROUDE GRID POINTS
089     C       THG = ARRAY OF FROUDE GRID POINT ELEVS (M MSL)
090     C       OH = ARRAY OF OBSTACLE HEIGHTS FOR EACH FROUDE GRID POINT (M)
091     C       THET = ARRAY OF DRAINAGE DIRECTION FOR EACH FROUDE GRID POINT
092     C              (DEG - 000 NORTH)
093     C       XGF = ARRAY OF X-COORD OF GRID POINTS (KM)
094     C       YGF = ARRAY OF Y-COORD OF GRID POINTS (KM)
095     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
096     C  CONSTANTS DEFINITIONS:
097     C       GRAV = 9.81   [ACCELERATION DUE TO GRAVITY (M/S**2)]
098     C       RPD = .017453   [RADIANS PER DEGREE]
099     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
100     C  FILES READ FROM:
101     C       LOGICAL UNIT 15 = SEQUENTIAL BINARY DISK FILE CONTAINING SPECS
102     C                         AND DATA OF FROUDE TERRAIN FILE, CREATED BY
103     C                         PROGRAM 'TERAIN'.
104     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
105     C  FILES WRITTEN TO:
106     C       LOGICAL UNIT LUO = IF DETLF IS .TRUE. THEN RUN INFO OUTPUT
107     C       LOGICAL UNIT  6 = ERROR MESSAGES
108     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
109     C  CALLING SUBROUTINES:
110     C       AMPLIT
111     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
112     C  CALLED SUBROUTINES:
113     C       AMPL, LAPSE, TEMP, TETRI, SMALLH, WINGAM, TPROF, WADJ, WNDRSP
114     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
115     C  COMMON BLOCKS:
116     C       MAIN2, AMPL1, FROUD
117     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
118     C  PARAMETERS DEFINITIONS:
119     C       NAL = NUA+NNS+NOT+NFR
120     C       NGA = NUMBER OF 'GAMMA' SURFACES
121     C       NPF = NUMBER OF GRID POINTS IN FROUDE NUMBER COMPUTATION
122     C       NPO = NUMBER OF POLYNOMIAL TERMS
123     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
124     C
```

```
      SUBROUTINE FROUDE (JUL,IHR,NFS,NGAMA,HTGAMA,NTERMS,NTERMA,RAD,
     +                   NSTA,NSSTA,IBTZ,FRO1ST,HA,ROA,IC,LUO)
C
      PARAMETER
     +    NAL=155,        NGA=9,          NPF=10,         NPO=21
C
      LOGICAL
     +    FRO1ST,         TMP1ST,         DETLF
C
C!!!  LOGICAL   ADJ
C
      DIMENSION
     +    HTGAMA(NGA),    RU(NGA),        RV(NGA),        RW(NGA),
     +    HDVD(NPF,NPF),  WDD(NGA),       WS(NGA),        WDI(NGA),
     +    WSI(NGA),       WDO(NGA),       WSO(NGA)
C
      COMMON /MAIN2/
     +    HH,             IHH,            IX0,            IY0,
     +    RXY,            SLX,            SLY
C
      COMMON /AMPL1/
     +    UP(NAL,NGA),    VP(NAL,NGA),    XG(NAL),        YG(NAL),
     +    ZSTER(NAL)
C
      COMMON /FROUD/
     +    AG(NPF),        BG(NPF),        OH(NPF,NPF),    THET(NPF,NPF),
     +    THG(NPF,NPF),   XGF(NPF),       YGF(NPF),       NXCF,
     +    NYCF
C
      GRAV=9.81
      RPD=.017453
      NTERM1=15
      NTERM2=21
      TMP1ST=.FALSE.
      DETLF=.FALSE.
C
C             ***READ IN REQUIRED TERRAIN STATISTICS FOR EACH GRID
C             POINT.
      IF (FRO1ST)  THEN
        FRO1ST=.FALSE.
        TMP1ST=.TRUE.
        READ (15) IX0F,IY0F,NXCF,NYCF,ICSF
        SSLX=REAL(NXCF*ICSF)
        SSLY=REAL(NYCF*ICSF)
        IF (IX0F.NE.IX0.OR.IY0F.NE.IY0.OR.SSLX.NE.SLX.OR.SSLY.NE.SLY)
     +  THEN
          WRITE (6,20)
          STOP 'FROUDE'
        ENDIF
C
        DO 1 M=1,NYCF
          READ (15) (THG(L,M),L=1,NXCF)
          READ (15) (OH(L,M),L=1,NXCF)
          READ (15) (THET(L,M),L=1,NXCF)
    1   CONTINUE
C
C             ***DETERMINE ALPHA AND BETA, AND X AND Y COORDINATES OF
C             ***GRID POINTS.
        XCF=REAL(NXCF)
        YCF=REAL(NYCF)
        RX0=REAL(IX0F)
        RY0=REAL(IY0F)
```

```
187             STX=1./XCF
188             AG(1)=STX/2.
189             XGF(1)=AG(1)*SSLX+RX0
190             DO 5 I=2,NXCF
191               AG(I)=AG(I-1)+STX
192               XGF(I)=AG(I)*SSLX+RX0
193           5 CONTINUE
194             STY=1./YCF
195             BG(1)=STY/2.
196             YGF(1)=BG(1)*SSLY+RY0
197             DO 6 J=2,NYCF
198               BG(J)=BG(J-1)+STY
199               YGF(J)=BG(J)*SSLY+RY0
200           6 CONTINUE
201     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
202             WRITE(LUO,39) NTERMS,NTERM1,NTERM2
203             WRITE(LUO,40)
204             WRITE(LUO,41) (XGF(L),L=1,NXCF)
205             DO 30 M=NYCF,1,-1
206               WRITE(LUO,42) YGF(M),(THG(L,M),L=1,NXCF)
207          30 CONTINUE
208             WRITE(LUO,43) (XGF(L),L=1,NXCF)
209             DO 31 M=NYCF,1,-1
210               WRITE(LUO,42) YGF(M),(OH(L,M),L=1,NXCF)
211          31 CONTINUE
212             WRITE(LUO,44) (XGF(L),L=1,NXCF)
213             DO 32 M=NYCF,1,-1
214               WRITE(LUO,42) YGF(M),(THET(L,M),L=1,NXCF)
215          32 CONTINUE
216     C*******************************************************************************
217           ENDIF
218     C
219     C               ***FILL AND UPDATE TEMPERATURE ARRAYS FOR COMPUTING
220     C               ***PTLR AND TMP.
221             CALL TEMP(JUL,IHR,NSTA,NSSTA,IBTZ,TMP1ST)
222     C
223     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
224           IF(DETLF) THEN
225             WRITE(LUO,45)
226           ENDIF
227     C*******************************************************************************
228     C
229     C               ***LOOP ON GRID POINTS
230           IC=NFS
231           DO 16 M=1,NYCF
232           DO 16 L=1,NXCF
233     C
234     C               ***COMPUTE VELOCITY COMPONENTS FOR ALL 'GAMMA' SURFACES
235     C               ***AT THE REQUESTED GRID POINT.
236           DZA=0.
237           DZB=0.
238           CALL WINGAM(AG(L),BG(M),THG(L,M),DZA,DZB,NTERMS,NGAMA,HTGAMA,
239          +            HA,ROA,HH,IHH,SLX,RXY,RU,RV,RW)
240     C
241     C               ***INTERPOLATE TEMPERATURE LAPSE RATE AND AVE TEMPERATURE
242     C               ***TO GRID POINT.
243           CALL LAPSE(JUL,IHR,L,M,RAD,PTLR,TMP)
244     C
245     C               ***COMPUTE 'SMALLH' AT GRID POINT
246           CALL SMALLH(IHH,HH,THG(L,M),AG(L),BG(M),HZS)
247     C
248     C               ***COMPUTE DIVIDING STREAMLINE HEIGHT FOR GRID POINT.
```

```
249           IF(PTLR.LE.0.) THEN
250             HDVD(L,M)=0.
251             GO TO 16
252           ELSE
253             OHG=OH(L,M)/HZS
254             DO 3 NG=1,NGAMA
255               WS(NG)=SQRT(RU(NG)**2+RV(NG)**2)
256         3   CONTINUE
257             CALL TPROF(NGAMA,HTGAMA,WS,OHG,WSPD)
258             FR=WSPD/(OH(L,M)*SQRT(GRAV*PTLR/TMP))
259             IF (FR.GE.1.) THEN
260               HDVD(L,M)=0.
261               GO TO 16
262             ELSE
263               HDVD(L,M)=OH(L,M)*(1.-FR)
264             ENDIF
265           ENDIF
266     C
267     C               ***RESOLVE DRAINAGE VECTOR INTO COMPONENTS.
268           ANG=(270.-THET(L,M))*RPD
269           DRX=-COS(ANG)
270           DRY=-SIN(ANG)
271     C
272     C               ***LOOP ON SURFACES ADJUSTING WINDS.
273     C!!!        ADJ=.FALSE.
274           IC=IC+1
275           DO 15 NG=1,NGAMA
276           HTAG=HTGAMA(NG)*HZS
277           IF (HTAG.GE.HDVD(L,M)) THEN
278             UP(IC,NG)=RU(NG)
279             VP(IC,NG)=RV(NG)
280             XG(IC)=AG(L)
281             YG(IC)=BG(M)
282             ZSTER(IC)=THG(L,M)
283           ELSE
284     C
285     C               ***MAKE WIND VECTOR TO UNIT VECTOR.
286             SP=(RU(NG)**2.+RV(NG)**2.)**.5
287             WX=RU(NG)/SP
288             WY=RV(NG)/SP
289     C
290     C               ***DETERMINE IF WIND BLOWING TOWARD OBSTACLE (IF AA
291     C               ***GREATER THAN 0. MEANS BLOWING TOWARD OBSTACLE).
292             AA=DRX*WX+DRY*WY
293             IF (AA.GT.0.)  THEN
294     C               **DETERMINE TANGENT WIND VECTOR.
295               TT=-DRY*WX+DRX*WY
296               IF (TT.GT.0.)  THEN
297                 TAX=-DRY
298                 TAY=DRX
299               ELSE
300                 TAX=DRY
301                 TAY=-DRX
302               ENDIF
303     C!!!        ADJ=.TRUE.
304               UP(IC,NG)=TAX*SP
305               VP(IC,NG)=TAY*SP
306               XG(IC)=AG(L)
307               YG(IC)=BG(M)
308               ZSTER(IC)=THG(L,M)
309     C
310             ELSE
```

```
311     C              ***DO NOT ADJUST WINDS SINCE WIND BLOWING AWAY FROM
312     C              ***OBSTACLE.
313               UP(IC,NG)=RU(NG)
314               VP(IC,NG)=RV(NG)
315               XG(IC)=AG(L)
316               YG(IC)=BG(M)
317               ZSTER(IC)=THG(L,M)
318             ENDIF
319           ENDIF
320        15 CONTINUE
321     C!!!       IF(.NOT.ADJ) THEN
322     C!!!         IC=IC-1
323     C!!!         GO TO 16
324     C!!!       ENDIF
325     C
326     C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
327           IF(DETLF) THEN
328     C               ***COMPUTE ADJUSTMENTS TO WINDS FOR EACH GRID POINT
329               DO 54 KK=1,NGAMA
330                 CALL WNDRSP(UP(IC,KK),VP(IC,KK),WDO(KK),WSO(KK))
331                 CALL WNDRSP(RU(KK),RV(KK),WDI(KK),WSI(KK))
332                 IF(WSI(KK).NE.0.) THEN
333                   WDD(KK)=WDO(KK)-WDI(KK)
334                   IF(WDD(KK).GE.270.) WDD(KK)=WDD(KK)-360.
335                   IF(WDD(KK).LE.-270.) WDD(KK)=WDD(KK)+360.
336                 ELSE
337                   WDD(KK)=999.
338                 ENDIF
339        54       CONTINUE
340               WRITE(LUO,46) XGF(L),YGF(M),(WDD(KK),KK=1,NGAMA)
341           ENDIF
342     C*******************************************************************************
343     C
344        16 CONTINUE
345     C
346     C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
347           IF(DETLF) THEN
348             WRITE(LUO,47) (XGF(L),L=1,NXCF)
349             DO 33 M=NYCF,1,-1
350               WRITE(LUO,42) YGF(M),(HDVD(L,M),L=1,NXCF)
351        33     CONTINUE
352           ENDIF
353     C*******************************************************************************
354     C
355     C               ***CHECK TO SEE IF ANY GRID POINTS HAVE BEEN MODIFIED -
356     C               ***IF SO COMPUTE NEW AMPLITUDE FUNCTIONS.
357           ICK=IC-NFS
358           IF (ICK.GT.0)  THEN
359             CALL WADJ (NFS,NGAMA,HTGAMA,HA,ROA,HH,IHH)
360             NTERMA=NTERMS
361             IF(ICK.GE.20) NTERMA=NTERM1
362             IF(ICK.GE.40) NTERMA=NTERM2
363             CALL AMPL (NTERMA,IC,NGAMA,HTGAMA,HA,ROA,HH,IHH,RXY)
364     C*****************WRITE OUT RUN INFO TO UNIT LUO*************************
365           IF(DETLF) THEN
366             WRITE(LUO,48) NTERMA
367             DO 65 K=NFS+1,IC
368     C             **COMPUTE OUTPUT WINDS ON ALL GAMMA SURFACES
369               CALL TETRI(XG(K),YG(K),TH,DZA,DZB)
370               CALL WINGAM(XG(K),YG(K),TH,DZA,DZB,NTERMA,NGAMA,HTGAMA,
371          +                HA,ROA,HH,IHH,SLX,RXY,RU,RV,RW)
372               DO 64 KK=1,NGAMA
```

```
373                 CALL WNDRSP(UP(K,KK),VP(K,KK),WDI(KK),WSI(KK))
374                 CALL WNDRSP(RU(KK),RV(KK),WDO(KK),WSO(KK))
375        64       CONTINUE
376                 GRX=XG(K)*SLX+REAL(IX0)
377                 GRY=YG(K)*SLY+REAL(IY0)
378                 WRITE(LUO,49) GRX,GRY,(WDI(KK),WSI(KK),KK=1,NGAMA)
379                 WRITE(LUO,50) GRX,GRY,(WDO(KK),WSO(KK),KK=1,NGAMA)
380        65     CONTINUE
381             ENDIF
382     C*************************************************************************
383             ELSE
384               NTERMA=0
385             ENDIF
386     C
387             RETURN
388            20 FORMAT (2X,'DOMAIN SPECS IN FROUDE # MODIFICATION DO NOT MATCH','
389              +SPECS IN MAIN PROGRAM.')
390            39 FORMAT(1H0,/,5X,'THE NUMBER OF POLYNOMIAL TERMS USED IN THE',
391              +' FROUDE NUMBER MODIFICATIONS',/,5X,'ARE ',I2,', ',I2,', OR ',I2,
392              +' DEPENDING ON THE NUMBER OF CONSTRUCTED "SOUNDINGS" REMAINING.')
393            40 FORMAT(1H ,/,5X,'THE TERRAIN STATISTICS USED IN THE FROUDE NUMBER'
394              +,' MODIFICATIONS ARE:')
395            41 FORMAT(1H ,/,7X,'LOWER TERRAIN HEIGHT (M MSL)',/,41X,'X (KM)',
396              +/,7X,'Y (KM)',1X,10(2X,F5.0),/)
397            42 FORMAT(1H ,7X,F4.0,2X,10(2X,F5.0))
398            43 FORMAT(1H ,/,7X,'OBSTACLE HEIGHT (M)',/,41X,'X (KM)',/,7X,
399              +'Y (KM)',1X,10(2X,F5.0),/)
400            44 FORMAT(1H ,/,7X,'DRAINAGE DIRECTION (DEGS FROM NORTH)',/,41X,
401              +'X (KM)',/,7X,'Y (KM)',1X,10(2X,F5.0),/)
402            45 FORMAT(1H ,/,7X,'THE FROUDE NUMBER WIND DIRECTION MODIFICATION TO'
403              +,' THE ORIGINAL COMPUTED WINDS AT THE GRID POINTS ARE ("+" IS',
404              +' CLOCKWISE) :',/,9X,'X (KM)',2X,'Y (KM)',4X,'SURFACE 1',2X,
405              +'SURFACE 2',2X,'SURFACE 3',2X,'SURFACE 4',2X,'SURFACE 5',2X,
406              +'SURFACE 6',2X,'SURFACE 7',2X,'SURFACE 8',2X,'SURFACE 9')
407            46 FORMAT(1H ,9X,F4.0,4X,F4.0,9(6X,F5.0))
408            47 FORMAT(1H ,/,9X,'THE DIVIDING STREAMLINE HEIGHTS ARE (M):',/,43X,
409              +'X (KM)',/,9X,'Y (KM)',1X,10(2X,F5.0),/)
410            48 FORMAT(1H ,/,7X,'THE ADJ WINDS AND COMPUTED WINDS AT THE GRID',
411              +' POINTS AFTER FROUDE MODS ARE ( NTERMS =',I3,' ):',/,31X,
412              +'SURFACE 1',2X,'SURFACE 2',2X,'SURFACE 3',2X,'SURFACE 4',2X,
413              +'SURFACE 5',2X,'SURFACE 6',2X,'SURFACE 7',2X,'SURFACE 8',2X,
414              +'SURFACE 9',/,9X,'I/O',2X,'X (KM)',2X,'Y (KM)',9(4X,'WD',3X,'WS'))
415            49 FORMAT(1H ,9X,'I',4X,F4.0,4X,F4.0,2X,9(2X,F4.0,1X,F4.1))
416            50 FORMAT(1H ,9X,'O',4X,F4.0,4X,F4.0,2X,9(2X,F4.0,1X,F4.1))
417               END
```

```
001     C  SUBROUTINE: MET.GRID
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO CONVERT FROM GRAMA GRID COORDINATES
005     C  IN KILOMETERS TO LATITUDE-LONGITUDE IN DEGREES.  EQUATIONS FOR A
006     C  TRANSVERSE MERCATOR PROJECTION ASSUMING THE EARTH IS SHAPED AS A
007     C  SPHERE ARE USED.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       COX = X-COORDINATE IN GRAMA COORDINATE SYSTEM (KM)
011     C       COY = Y-COORDINATE IN GRAMA COORDINTAE SYSTEM (KM)
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C       XLAT = LATITUDE OF POINT (DEG)
015     C       YLONG = LONGITUDE OF POINT (DEG)
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  CONSTANT DEFINITIONS:
018     C       X0 = 250.5  [DIST. OF GRAMA ORIGIN WEST OF CM (KM)]
019     C       Y0 = 4224.697  [DIST. IF GRAMA ORIGIN NORTH OF EQUATOR (KM)]
020     C       R = 6367.3858  [MEAN RADIUS OF EARTH (KM)]
021     C       C = .9996  [PROJECTION SCALING FACTOR]
022     C       CM = 109.  [CENTRAL MERIDIAN OF TRANS. MERCATOR PROJ. (DEG)]
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLING SUBROUTINES:
025     C       BLPARS, COEFF, MIXSTB, SRFC, UPPER
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLED SUBROUTINES:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  COMMON BLOCKS:
031     C       (NONE)
032     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
033     C
034           SUBROUTINE GRID(COX,COY,XLAT,YLONG)
035     C
036           DOUBLE PRECISION B,XPRIM,YPRIM,A,D
037     C
038     C                ***CONSTANTS
039           X0=250.5
040           Y0=4224.697
041           R=6367.3858
042           C=.9996
043           CM=109.
044           CO1=180./3.141592654
045           CO2=3.141592654/2.
046     C
047           B=EXP(2.*(COX-X0)/(C*R))
048           XPRIM=(B-1.)/(B+1.)
049           YPRIM=TAN(CO2-(COY+Y0)/(C*R))
050           A=DSQRT((1.-XPRIM**2)/(1.+YPRIM**2))
051           D=DACOS(YPRIM*A/DSQRT(1.-A**2))*CO1
052     C
053     C                ***COMPUTE LATITUDE
054           XLAT=DASIN(A)*CO1
055     C
056     C                ***COMPUTE LONGITUDE
057           IF(XPRIM.LT.0.) THEN
058             YLONG=CM+D
059           ELSE
060             YLONG=CM-D
061           ENDIF
062     C
```

```
063             RETURN
064             END
```

```
001     C  SUBROUTINE: MET.HGTS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE UPPER AIR SOUNDING
005     C  HEIGHTS IN (M MSL) GIVEN HEIGHTS IN (MB).
006     C
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C  THE INPUTS ARE:
009     C       DP = ARRAY OF DEW POINT TEMP. OBS. IN SND. (DEG C)
010     C       HTSTA = HEIGHT OF STATION IN (M MSL)
011     C       NOBT = NUMBER OF OBSERVATIONS IN SOUNDING
012     C       THTP = ARRAY OF OBS. HEIGHTS IN SND. (MB)
013     C       TMP = ARRAY OF TEMP. OBS. IN SND. (DEG C)
014     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
015     C  THE OUTPUTS ARE:
016     C       THTM = ARRAY OF OBS. HEIGHTS IN SND. (M MSL)
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  CONSTANT DEFINITIONS:
019     C       C = 0.034075  [(ACCEL. DUE TO GRAV)/UNIV. GAS CONST.)]
020     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
021     C  CALLING SUBROUTINES:
022     C       UPPER
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLED SUBROUTINES:
025     C       (NONE)
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  COMMON BLOCKS:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  PARAMETER DEFINITIONS:
031     C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
032     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
033     C
034           SUBROUTINE HGTS (NOBT,THTP,THTM,TMP,DP,HTSTA)
035     C
036           PARAMETER
037          +    NOB=30
038           DIMENSION
039          +    DP(NOB),       THTM(NOB),      THTP(NOB),      TMP(NOB),
040          +    W(NOB)
041     C
042           C=0.034075
043           DO 2 I=1,NOBT
044     C              ***CALCULATE ACTUAL VAPOR PRESSURE (E), MIXING RATIO (W),
045     C              ***AND VIRTUAL TEMPERATURE (TSTAR).
046             E=6.1078*10.**(7.5*DP(I)/(237.3+DP(I)))
047             W(I)=.622*E/(THTP(I)-E)
048             IF (I.EQ.1) GO TO 1
049             TSTAR1=(TMP(I)+273.15)*(1.+1.609*W(I))/(1.+W(I))
050             TSTAR2=(TMP(I-1)+273.15)*(1.+1.609*W(I-1))/(1.+W(I-1))
051             TSTAR=(TSTAR1+TSTAR2)/2.
052             DELTAH=(TSTAR/C)*ALOG(THTP(I-1)/THTP(I))
053             THTM(I)=THTM(I-1)+DELTAH
054             GO TO 2
055           1 THTM(I)=0.
056           2 CONTINUE
057     C
058           DO 3 I=1,NOBT
059             THTM(I)=THTM(I)+HTSTA
060           3 CONTINUE
061     C
062           RETURN
```

```
063          END
```

```
001     C  SUBROUTINE: MET.HPROF
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO DETERMINE THE HEIGHT OF THE
005     C  INTERSECTION OF A POTENTIAL TEMPERATURE VALUE WITH THE MORNING
006     C  POTENTIAL TEMPERATURE SOUNDING.  IF THE INPUT POT. TEMP. VALUE
007     C  IS LESS THEN THE SURFACE POT. TEMP. OBS. THEN THE HEIGHT IS SET
008     C  EQUAL TO THE SURFACE HEIGHT.  IF THE POT. TEMP. VALUE DOES NOT
009     C  INTERSECT THE SOUNDING THEN THE HEIGHT IS SET EQUAL TO THE
010     C  HEIGHT OF THE MAX. POT. TEMP. OBSERVATION.
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE INPUTS ARE:
013     C       NOBS = NUMBER OF OBSERVATIONS IN SOUNDING.
014     C       THT = ARRAY OF OBSERVATION HEIGHTS (M MSL)  [OR M AGL].
015     C       PT = ARRAY OF POTENTIAL TEMPERATURES (DEG K).
016     C       SPT = POT. TEMP. VALUE AT WHICH TO COMPUTE HGHT. (DEG K).
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       DH = HEIGHT AT WHICH POT. TMP. OBS. INTERSECTS MORNING
020     C            SOUNDING (M MSL)  [OR M AGL].
021     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
022     C  CALLING SUBROUTINES:
023     C       LAPSE, MIXSTB
024     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
025     C  CALLED SUBROUTINES:
026     C       (NONE)
027     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
028     C  PARAMETER DEFINITIONS:
029     C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
030     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
031     C
032           SUBROUTINE HPROF (NOBS,THT,PT,SPT,DH)
033     C
034           PARAMETER
035          +    NOB=30
036           DIMENSION
037          +    PT(NOB),       THT(NOB)
038     C
039           IF (NOBS.LE.1)  THEN
040             WRITE (6,3)
041             STOP 'HPROF'
042           ENDIF
043     C
044     C              ***DETERMINE HEIGHT AT WHICH SURFACE OBS INTERSECTS
045     C              ***SOUNDING.
046           IF (SPT.LE.PT(1))  THEN
047             DH=THT(1)
048             RETURN
049           ENDIF
050     C
051           DO 2 J=2,NOBS
052             IF (SPT.LE.PT(J))  THEN
053               SLP=(PT(J)-PT(J-1))/(THT(J)-THT(J-1))
054               CPT=PT(J)-SLP*THT(J)
055               DH=(SPT-CPT)/SLP
056               RETURN
057             ENDIF
058         2 CONTINUE
059     C
060     C              ***SINCE SURFACE POTENTIAL TEMPERATURE DID NOT INTERSECT
061     C              ***SOUNDING THEN SET DH TO HEIGHT OF MAXIMUM TEMPERATURE
062     C              ***ON SOUNDING
```

```
063           DH=THT(1)
064           DO 4 J=2,NOBS
065             IF(PT(J).GT.PT(J-1)) DH=THT(J)
066         4 CONTINUE
067     C
068         3 FORMAT (2X,'NEED MORE THAN ONE OBSERVATION IN PROFILE.')
069           END
```

```
001     C  SUBROUTINE: MET.JULIAN
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE JULIAN DATE AND HOUR-OF-
005     C  DAY IN THE BASE TIME ZONE GIVEN AN 8 DIGET INTEGER NUMBER
006     C  CODED WITH THE MONTH, DAY, YEAR, AND HOUR, AND THE TIME ZONE
007     C  OF THE CODED TIME.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       ITIME = 8 DIGET INTEGER NUMBER CODED WITH 'MO'DA'YR'HR'
011     C       ITZO = TIME ZONE OF CODED TIME = 0(ZULU), 5(EASTERN),
012     C                        6(CENTRAL), 7(MOUNTAIN), 8(PACIFIC).
013     C       IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL),
014     C                               7(MOUNTAIN), 8(PACIFIC).
015     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
016     C  THE OUTPUTS ARE:
017     C       IHR = HOUR-OF-DAY (01 TO 24)
018     C       JDATE = JULIAN DATE (1 TO 366)
019     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
020     C  CALLING SUBROUTINES:
021     C       AAMAIN, SRFC, UPPER
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLED SUBROUTINES:
024     C       (NONE)
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  COMMON BLOCKS:
027     C       (NONE)
028     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
029     C
030             SUBROUTINE JULIAN (ITIME,ITZO,IBTZ,IHR,JDATE)
031     C
032             DIMENSION
033            +    NDAY(12)
034     C
035             DATA NDAY/0,31,59,90,120,151,181,212,243,273,304,334/
036     C
037             TIM=REAL(ITIME)
038             MO=INT(TIM/10.**6.)
039             IDAY=INT(TIM/10.**4.)-MO*10**2
040             IYR=INT(TIM/10.**2.)-MO*10**4-IDAY*10**2
041             IHR=ITIME-INT(TIM/10.**2.)*10**2
042     C
043     C                ***CHECK FOR MONTH INDEX IN RANGE.
044             IF (MO.LE.0.OR.MO.GE.13)  THEN
045               WRITE (6,1)
046               STOP 'JULIAN'
047             ENDIF
048     C
049     C                ***CONVERT TO JULIAN DATE.
050             JDATE=IDAY+NDAY(MO)
051     C
052     C                ***ADJUST FOR LEAP YEAR.
053             A=FLOAT(IYR)/4-IYR/4
054             IF (A.EQ.0..AND.MO.GE.3) JDATE=JDATE+1
055     C
056     C                ***CONVERT TO BASE TIME ZONE
057             IHR=IHR+(ITZO-IBTZ)
058             IF(IHR.LE.0) THEN
059               IHR=IHR+24
060               JDATE=JDATE-1
061             ENDIF
062             IF(IHR.GT.24) THEN
```

```
063             IHR=IHR-24
064             JDATE=JDATE+1
065           ENDIF
066     C
067           RETURN
068         1 FORMAT (5X,'MONTH INDEX IS OUT OF RANGE.')
069           END
```

```
001     C  SUBROUTINE: MET.LAPSE
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE POTENTIAL TEMPERATURE
005     C  LAPSE RATE AND THE BULK TEMPERATURE FOR AN OBSTACLE HEIGHT
006     C  USED IN THE FROUDE NUMBER FLOW MODIFICATION SCHEME.  THE LAPSE
007     C  RATE AND TEMPERATURE ARE COMPUTED AT A GRID POINT FOR A GIVEN HOUR
008     C  OF THE SIMULATION.  THE SIMULATION MUST START BETWEEN SUNRISE AND
009     C  SUNSET.  THE LAPSE RATE AND AVE TEMPERATURE ARE COMPUTED AT A GRID
010     C  POINT BY '1/R**2' WEIGHTING THE LAPSE RATE AND AVE TEMPERATURE FROM
011     C  WEATHER STATIONS TO THE GRID POINTS (REFER TO THE DOCUMENT ON THE
012     C  'MET' PROGRAM).
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  THE INPUTS ARE:
015     C    THRU ARGUMENT:
016     C      JUL = JULIAN DATE OF DAY BEING PROCESSES
017     C      IHRR = HOUR OF DAY BEING PROCESSED (1 TO 24)
018     C      I = HORIZONTAL INDEX OF GRID POINT
019     C      J = VERTICAL INDEX OF GRID POINT
020     C      RAD = MAX. DISTANCE GREATER THAN WHICH DATA WILL NOT BE
021     C            INTERPOLATED TO A GRID POINT
022     C    THRU COMMON BLOCK 'UPPER1':
023     C      COX = X-COORD OF UPPER AIR STATION LOCATION IN GRAMA COORD (KM)
024     C      COY = Y-COORD OF UPPER AIR STATION LOCATION IN GRAMA COORD (KM)
025     C      EL = ELEVATION OF UPPER AIR STATIONS (M MSL)
026     C    THRU COMMON BLOCK 'TEMP1':
027     C      IU = ARRAY OF INDEX NUMBERS OF UPPER AIR STATIONS USED
028     C      CNGTIM = ARRAY OF A LOGICAL VARIABLE FOR EACH STATION IF .TRUE.
029     C               INDICATING THAT THE CURRENT SIMULATION HOUR (IHRR) IS
030     C               ON DAY 'JUL+1' BUT BEFORE SUNRISE.  THEREFORE 24 MUST
031     C               SUBTRACTED FROM THE HOUR-OF-DAY ARRAY INDEX IN THE
032     C               SURFACE DATA ARRAYS TO GET THE CORRECT HOUR.
033     C      DAYFL = ARRAY OF A LOGICAL VARIABLE FOR EACH STATION IF .TRUE.
034     C               INDICATING DATA AVAILABLE FROM SUNRISE TO SUNSET ON
035     C               DAY 'JUL'.
036     C      NTFL = ARRAY OF A LOGICAL VARIABLE FOR EACH STATION IF .TRUE.
037     C             INDICATING DATA AVAILABLE FROM SUNSET ON DAY 'JUL' TO
038     C             SUNRISE ON DAY 'JUL+1'.
039     C    THRU COMMON BLOCK 'LAPSE1':
040     C      ISRS = ARRAY OF SUNRISE TIME FOR EACH STATION ON DAY 'JUL'.
041     C      ISSS = ARRAY OF SUNSET TIME FOR EACH STATION ON DAY 'JUL'.
042     C      NTS = NUMBER OF STATIONS WITH BOTH UPPER AIR AND SURFACE DATA.
043     C      NOBM = ARRAY OF NUM. OF OBS. FOR MORN. SND. FOR STATIONS
044     C      THTM = ARRAY OF OBS. HEIGHTS FOR MORN. SND. FOR STATIONS (M MSL)
045     C      TMPM = ARRAY OF TEMP. OBS. FOR MORN. SND. FOR STATIONS (DEG C)
046     C      PTM = ARRAY OF POT. TEMP. OBS. FOR MORN SND FOR STATNS (DEG K)
047     C      NOBA = -------I
048     C      THTA =        I----  SAME AS ABOVE EXCEPT FOR AFTERNOON
049     C      TMPA =        I      SOUNDING ON DAY 'JUL'
050     C      PTA = --------I
051     C      NOBM1 = --------I
052     C      THTM1 =         I----  SAME AS ABOVE EXCEPT FOR MORNING
053     C      TMPM1 =         I      SOUNDING ON DAY 'JUL+1'
054     C      PTM1 = ---------I
055     C      NOBMM = ----I      SAME AS MORNING SOUNDING ON DAY 'JUL' GIVEN
056     C      THTMM =     I----  ABOVE EXCEPT LOWER PART OF POT TEMP SND HAS
057     C      PTMM = -----I      BEEN MODIFIED TO MATCH ADJUSTED SRFC POT TMP
058     C      APT = ARRAY OF HOURLY ADJUSTED SRFC POT TEMP VALUES STARTING AT
059     C            SUNRISE ON DAY 'JUL' AND GOING TO SUNRISE ON DAY 'JUL+1'
060     C      ATMP = ARRAY OF HOURLY SRFC TEMP VALUES STARTING AT SUNRISE ON
061     C             DAY 'JUL' AND GOING TO SUNRISE ON DAY 'JUL+1'
062     C    THRU COMMON BLOCK 'FROUD':
```

```
C       XGF = ARRAY OF X-COORDS OF GRID POINTS IN GRAMA COORD (KM)
C       YGF = ARRAY OF Y-COORDS OF GRID POINTS IN GRAMA COORD (KM)
C       THG = ARRAY OF GRID POINT ELEVATIONS (M MSL)
C       OH = ARRAY OF OBSTACLE HEIGHTS AT GRID POINT (M)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C    THRU ARGUMENT:
C       DTDZ = POT TEMP LAPSE RATE THRU OBSTACLE DEPTH AT GRID POINT
C              I,J (DEGK/M)
C       BT = AVERAGE TEMP THRU OBSTACLE DEPTH AT GRID POINT I,J (DEGK)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  FILES WRITTEN TO:
C       LOGICAL UNIT 6 = ERROR MESSAGES
C**********************************************************************
C  CALLING SUBROUTINES:
C       FROUDE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C       HPROF, TPROF
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C       UPPER1, TEMP1, LAPSE1, FROUD
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C       NPF = NUMBER OF GRID POINTS IS FROUDE NUMBER COMPUTATION
C       NUA = NUMBER OF UPPER AIR STATIONS
C       NOB = NUMBER OF OBSERVATIONS IN AN UPPER AIR SOUNDING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE LAPSE(JUL,IHRR,I,J,RAD,DTDZ,BT)
C
      PARAMETER
     +    NOB=30,          NUA=10,          NPF=10
C
      LOGICAL
     +    YESSTA,          CNGTIM,          DAYFL,           NTFL
C
      CHARACTER*10
     +    NAME
C
      COMMON /UPPER1/
     +    COX(NUA),        COY(NUA),        EL(NUA),         IDU(NUA),
     +    NAME(NUA),       NSDT(NUA),       NSDW(NUA),       XLAT(NUA),
     +    YLONG(NUA),      IUSRTW(NUA),     IUSRTT(NUA),     IUENDW(NUA),
     +    IUENDT(NUA),     ITZU(NUA)
C
      COMMON /TEMP1/
     +    IS(NUA),         ISENDR(NUA),     ISPNT(NUA),      IU(NUA),
     +    IUENDR(NUA),     IUPNT(NUA),      JULI(NUA),       CNGTIM(NUA),
     +    DAYFL(NUA),      NTFL(NUA)
C
      COMMON /LAPSE1/
     +    APT(36,NUA),     ATMP(36,NUA),    DTDZCK(NUA),     ISRS(NUA),
     +    ISSS(NUA),       NOBA(NUA),       NOBM(NUA),       NOBMM(NUA),
     +    NOBM1(NUA),      NTS,             PTA(NOB,NUA),    PTM(NOB,NUA),
     +    PTMM(NOB,NUA),   PTM1(NOB,NUA),   THTA(NOB,NUA),   THTAP(NOB,NUA),
     +    THTM(NOB,NUA),   THTMM(NOB,NUA),  THTM1(NOB,NUA),  TMPA(NOB,NUA),
     +    TMPM(NOB,NUA),   TMPM1(NOB,NUA)
C
      COMMON /FROUD/
     +    AG(NPF),         BG(NPF),         OH(NPF,NPF),     THET(NPF,NPF),
     +    THG(NPF,NPF),    XGF(NPF),        YGF(NPF),        NXCF,
```

```
     +    NYCF
C
      DIMENSION
     +    HTS1(NOB),     VAL1(NOB),     HTS2(NOB),     VAL2(NOB)
C
      OHAGL=OH(I,J)
      HLL=THG(I,J)
      HLU=THG(I,J)+OH(I,J)
      WFF=0.
      DTDZS=0.
      BTS=0.
C
C
C               ****LOOP ON STATIONS
      DO 50 L=1,NTS
      OHMSLP=OHAGL+EL(IU(L))
      YESSTA=.FALSE.
      IF(CNGTIM(L)) IHR=IHRR+24
      IF(.NOT.CNGTIM(L)) IHR=IHRR
C
C               ***DAYTIME COMPUTATION
      IF(IHR.LE.ISSS(L).AND.DAYFL(L)) THEN
        YESSTA=.TRUE.
C               **COMPUTE DTDZ AT STATION
        IF(IHR.LT.ISSS(L)-2) THEN
          DO 1 II=1,NOBMM(L)
          HTS1(II)=THTMM(II,L)
          VAL1(II)=PTMM(II,L)
    1     CONTINUE
          CALL HPROF(NOBMM(L),HTS1,VAL1,APT(IHR,L),CMHMSL)
          CMHAGL=CMHMSL-EL(IU(L))
C               *FIND LOWER DTDZ
          IF(OHAGL.LE.CMHAGL) THEN
            DTDZL=0.
          ELSE
            CALL TPROF(NOBMM(L),HTS1,VAL1,OHMSLP,T)
            DTDZL=(T-APT(IHR,L))/OHAGL
          ENDIF
          IF(HLL.GT.EL(IU(L)).AND.HLU.GT.CMHMSL) THEN
C               *FIND UPPER DTDZ
            CALL TPROF(NOBMM(L),HTS1,VAL1,HLU,TU)
            IF(HLL.GT.CMHMSL) THEN
              CALL TPROF(NOBMM(L),HTS1,VAL1,HLL,TL)
              DTDZU=(TU-TL)/OHAGL
            ELSE
              DTDZU=(TU-APT(IHR,L))/OHAGL
            ENDIF
          ELSE
            DTDZU=DTDZL
          ENDIF
          DTDZ=(DTDZL+DTDZU)/2.
          IF(IHR.EQ.ISSS(L)-3) DTDZCK(L)=DTDZ
C               **FORCE DTDZ TO CONVERGE TO AFTERNOON SOUNDING.
        ELSE
C               *COMPUTE AFTERNOON SOUNDING LAPSE RATE
          DO 8 II=1,NOBA(L)
          HTS2(II)=THTA(II,L)
          VAL2(II)=PTA(II,L)
    8     CONTINUE
C               *FIND LOWER DTDZ
          CALL TPROF(NOBA(L),HTS2,VAL2,OHMSLP,T1)
          DTDZL=(T1-VAL2(1))/OHAGL
```

```
187                IF(HLL.GT.EL(IU(L))) THEN
188     C                 *FIND UPPER DTDZ
189                  CALL TPROF(NOBA(L),HTS2,VAL2,HLU,T2)
190                  CALL TPROF(NOBA(L),HTS2,VAL2,HLL,T1)
191                  DTDZU=(TU-TL)/OHAGL
192                ELSE
193                  DTDZU=DTDZL
194                ENDIF
195                DTDZA=(DTDZL+DTDZU)/2.
196                WF=FLOAT(IHR-(ISSS(L)-3))/3.
197                DTDZ=DTDZCK(L)+(DTDZA-DTDZCK(L))*WF
198              ENDIF
199     C                **COMPUTE AVERAGE TEMPERATURE THRU LAYER
200              DO 4 II=1,NOBM(L)
201              HTS1(II)=THTM(II,L)
202              VAL1(II)=TMPM(II,L)
203            4 CONTINUE
204              DO 5 II=1,NOBA(L)
205              HTS2(II)=THTA(II,L)
206              VAL2(II)=TMPA(II,L)
207            5 CONTINUE
208              WF=ABS((ATMP(IHR,L)-ATMP(ISRS(L),L))/(ATMP(ISSS(L),L)-
209             + ATMP(ISRS(L),L)))
210     C                *FIND LOWER BULK TEMPERATURE
211              CALL TPROF(NOBM(L),HTS1,VAL1,OHMSLP,T1)
212              CALL TPROF(NOBA(L),HTS2,VAL2,OHMSLP,T2)
213              TL=T1+(T2-T1)*WF
214              BTL=(TL+ATMP(IHR,L))/2.
215              IF(HLL.GT.EL(IU(L))) THEN
216     C                *FIND UPPER BULK TEMPERATURE
217                CALL TPROF(NOBM(L),HTS1,VAL1,HLU,T2)
218                CALL TPROF(NOBA(L),HTS2,VAL2,HLU,T1)
219                TU=T1+(T2-T1)*WF
220                CALL TPROF(NOBM(L),HTS1,VAL1,HLL,T2)
221                CALL TPROF(NOBA(L),HTS2,VAL2,HLL,T1)
222                TL=T1+(T2-T1)*WF
223                BTU=(TU-TL)/2.
224              ELSE
225                BTU=BTL
226              ENDIF
227              BT=(BTU+BTL)/2.
228            ENDIF
229     C
230     C                ***NIGHTIME COMPUTATION
231            IF(IHR.GT.ISSS(L).AND.NTFL(L)) THEN
232              YESSTA=.TRUE.
233     C                **COMPUTE DTDZ
234              DO 2 II=1,NOBM1(L)
235              HTS1(II)=THTM1(II,L)
236              VAL1(II)=PTM1(II,L)
237            2 CONTINUE
238              DO 3 II=1,NOBA(L)
239              HTS2(II)=THTA(II,L)
240              VAL2(II)=PTA(II,L)
241            3 CONTINUE
242              WF=ABS((APT(IHR,L)-APT(ISSS(L),L))/(APT(ISRS(L)+24,L)-
243             +    APT(ISSS(L),L)))
244     C                *FIND LOWER DTDZ
245              CALL TPROF(NOBM1(L),HTS1,VAL1,OHMSLP,T2)
246              CALL TPROF(NOBA(L),HTS2,VAL2,OHMSLP,T1)
247              TU=T1+(T2-T1)*WF
248              DTDZL=(TU-APT(IHR,L))/OHAGL
```

```
249             IF(HLL.GT.EL(IU(L))) THEN
250      C                *FIND UPPER DTDZ
251               CALL TPROF(NOBM1(L),HTS1,VAL1,HLU,T2)
252               CALL TPROF(NOBA(L),HTS2,VAL2,HLU,T1)
253               TU=T1+(T2-T1)*WF
254               CALL TPROF(NOBM1(L),HTS1,VAL1,HLL,T2)
255               CALL TPROF(NOBA(L),HTS2,VAL2,HLL,T1)
256               TL=T1+(T2-T1)*WF
257               DTDZU=(TU-TL)/OHAGL
258             ELSE
259               DTDZU=DTDZL
260             ENDIF
261             DTDZ=(DTDZL+DTDZU)/2.
262      C
263      C                **COMPUTE AVE TEMPERATURE THRU LAYER
264             DO 6 II=1,NOBM1(L)
265             HTS1(II)=THTM1(II,L)
266             VAL1(II)=TMPM1(II,L)
267           6 CONTINUE
268             DO 7 II=1,NOBA(L)
269             HTS2(II)=THTA(II,L)
270             VAL2(II)=TMPA(II,L)
271           7 CONTINUE
272             WF=ABS((ATMP(IHR,L)-ATMP(ISSS(L),L))/(ATMP(ISRS(L)+24,L)-
273            +ATMP(ISSS(L),L)))
274      C              *FIND LOWER BULK TEMPERATURE
275             CALL TPROF(NOBM1(L),HTS1,VAL1,OHMSLP,T2)
276             CALL TPROF(NOBA(L),HTS2,VAL2,OHMSLP,T1)
277             TL=T1+(T2-T1)*WF
278             BTL=(TL+ATMP(IHR,L))/2.
279             IF(HLL.GT.EL(IU(L))) THEN
280      C                *FIND UPPER BULK TEMPERATURE
281               CALL TPROF(NOBM1(L),HTS1,VAL1,HLU,T2)
282               CALL TPROF(NOBA(L),HTS2,VAL2,HLU,T1)
283               TU=T1+(T2-T1)*WF
284               CALL TPROF(NOBM1(L),HTS1,VAL1,HLL,T2)
285               CALL TPROF(NOBA(L),HTS2,VAL2,HLL,T1)
286               TL=T1+(T2-T1)*WF
287               BTU=(TU-TL)/2.
288             ELSE
289               BTU=BTL
290             ENDIF
291             BT=(BTU+BTL)/2.
292           ENDIF
293      C
294           IF(.NOT.YESSTA) GO TO 50
295      C
296      C                ***1/R**2 WEIGHT DTDZ AND BT TO GRID POINT.
297           RDIF=SQRT((COX(IU(L))-XGF(I))**2+(COY(IU(L))-YGF(J))**2)
298           IF(RDIF.LE.RAD) THEN
299             IF(RDIF.LE..1) RDIF=1.*10.**(-3.)
300             AINS=1./RDIF**2
301             WFF=AINS+WFF
302             DTDZS=AINS*DTDZ+DTDZS
303             BTS=AINS*BT+BTS
304           ENDIF
305      C
306        50 CONTINUE
307           IF(WFF.EQ.0.) THEN
308             WRITE(6,60) IHR,JUL
309             STOP 'LAPSE'
310           ENDIF
```

```
311           DTDZ=DTDZS/WFF
312           BT=BTS/WFF+273.15
313     C
314     C
315           RETURN
316        60 FORMAT(1H0,5X,'NO STATIONS FOR COMPUTATION OF DTDZ AND AVE TEMP',
317          +' AT HOUR',I4,'  ON JULIAN DAY',I5,' .')
318           END
```

```
001     C  SUBROUTINE: MET.LAT
002     C  DATE: APRIL  1985
003     C
004     C    THIS SUBROUTINE IS USED TO CONVERT FROM LATITUDE-LONGITUDE
005     C  IN DEGREES TO GRAMA COORDINATES IN KILOMETERS.  EQUATIONS FOR
006     C  A TRANSVERSE MERCATOR PROJECTION ASSUMING THE EARTH IS SHAPED
007     C  AS A SPHERE ARE USED.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       XLAT = LATITUDE OF POINT (DEG)
011     C       YLONG = LONGITUDE OF POINT (DEG)
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C       COX = X-COORDINATE IN GRAMA COORDINATE SYSTEM (KM)
015     C       COY = Y-COORDINATE IN GRAMA COORDINTAE SYSTEM (KM)
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  CONSTANT DEFINITIONS:
018     C       OX = 250.5  [DIST. OF GRAMA ORIGIN WEST OF CM (KM)]
019     C       OY = 4224.697  [DIST. IF GRAMA ORIGIN NORTH OF EQUATOR (KM)]
020     C       RAD = 6367.3858  [MEAN RADIUS OF EARTH (KM)]
021     C       C = .9996  [PROJECTION SCALING FACTOR]
022     C       CM = 109.  [CENTRAL MERIDIAN OF TRANS. MERCATOR PROJ. (DEG)]
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLING SUBROUTINES:
025     C       AMPLIT, COEFF, SRFC, UPPER, UTM
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLED SUBROUTINES:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  COMMON BLOCKS:
031     C       (NONE)
032     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
033     C
034           SUBROUTINE LAT (XLAT,YLONG,COX,COY)
035     C
036           DOUBLE PRECISION A,B
037     C
038     C                ***DEFINE CONSTANT PARAMETERS.
039           CO1=3.141592654/180.
040           CO2=3.141592654/2.
041           RAD=6367.3858
042           CM=109.*CO1
043           OX=250.5
044           OY=4224.697
045           C=.9996
046     C
047     C                ***COMPUTE X COORINATE.
048           XLATR=XLAT*CO1
049           YLONGR=YLONG*CO1
050           A=COS(XLATR)*SIN(CM-YLONGR)
051           COX=.5*C*RAD*DLOG((1.+A)/(1.-A))+OX
052     C
053     C                ***COMPUTE Y COORDINATE.
054           B=1./TAN(XLATR)*COS(CM-YLONGR)
055           COY=C*RAD*(CO2-DATAN(B))-OY
056     C
057           RETURN
058           END
```

```
001     C  SUBROUTINE: MET.LAY
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO LAYER AVERAGE OBSERVATIONS
005     C  GIVEN IN UPPER AIR SOUNDINGS.  THE ONLY CONSTRAINT IS
006     C  THAT THE HEIGHT OF THE LAST OBSERVATION BE GREATER THAN THE
007     C  HEIGHT OF THE TOP LAYER.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       NHTS = NUMBER OF OBSERVATIONS IN SOUNDING .
011     C       WHT = ARRAY OF OBSERVATION HEIGHTS (M MSL)  [OR M AGL].
012     C       WU = ARRAY OF INPUT OBSERVATIONS (E.G., WINDS (M/S), TEMP.
013     C            (DEG C), OR PRESSURE (MB)).
014     C       NLAYS = NUMBER OF LAYERS TO BE AVERAGED.
015     C       TOPL = ARRAY OF HGTS OF TOP OF EACH LAYER (M MSL)  [OR M AGL].
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  THE OUTPUTS ARE:
018     C       WOU = ARRAY OF OUTPUT LAYER-AVERAGED VALUES (E.G., WINDS (M/S),
019     C             TEMP, (DEG. C), OR PRESSURE (MB)).
020     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
021     C  CALLING SUBROUTINES:
022     C       UPPER, VALGAM
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLED SUBROUTINES:
025     C       TPROF
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  COMMON BLOCKS:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  PARAMETER DEFINITIONS:
031     C       NGA = NUMBER OF GAMMA SURFACES IN WIND FIELD COMPUTATIONS.
032     C             (THIS IS A MAXIMUM NUMBER OF LAYERS.)
033     C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
034     C       NTO = NGA + NOB
035     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
036     C
037             SUBROUTINE LAY (NHTS,WHT,WU,WOU,NLAYS,TOPL)
038     C
039             PARAMETER
040            +     NGA=9,          NOB=30,     NTO=39
041     C
042             DIMENSION
043            +     GU(NGA),        HT(NTO),    TOPL(NGA),     U(NTO),
044            +     WHT(NOB),       WHTN(NTO),  WOU(NGA),      WU(NOB),
045            +     WUN(NTO)
046     C
047     C               ***DETERMINE VALUES FOR LAYER TOPS.
048             DO 4 K=1,NLAYS
049               CALL TPROF(NHTS,WHT,WU,TOPL(K),GU(K))
050           4 CONTINUE
051     C
052     C               ***ORDER HEIGHT OF LAYER TOPS AND INFLECTION POINTS IN
053     C               ***ASCENDING ORDER IN A NEW ARRAY.
054             NNHTS=0
055             DO 6 J=1,NHTS
056               DO 5 K=1,NLAYS
057                 IF (TOPL(K).EQ.WHT(J)) GO TO 6
058           5   CONTINUE
059               NNHTS=NNHTS+1
060               WHTN(NNHTS)=WHT(J)
061               WUN(NNHTS)=WU(J)
062           6 CONTINUE
```

```
063     C
064           IS=1
065           L=0
066           DO 9 K=1,NLAYS
067             DO 7 J=IS,NNHTS
068               IF (TOPL(K).LT.WHTN(J)) GO TO 8
069               L=L+1
070               U(L)=WUN(J)
071               HT(L)=WHTN(J)
072         7     CONTINUE
073         8     CONTINUE
074               L=L+1
075               HT(L)=TOPL(K)
076               U(L)=GU(K)
077               IS=J
078         9   CONTINUE
079     C
080     C                ***COMPUTE AREA UNDER LINE SEGMENTS AND DETERMINE LAYER
081     C                ***AVERAGED WINDS.
082           IS=2
083           DO 12 K=1,NLAYS
084             AU=0.
085             DO 10 J=IS,L
086               AU=0.5*(U(J)+U(J-1))*(HT(J)-HT(J-1))+AU
087               IF (HT(J).EQ.TOPL(K)) GO TO 11
088        10     CONTINUE
089        11     CONTINUE
090               IF (K.EQ.1) WOU(K)=AU/(TOPL(K)-WHT(1))
091               IF (K.NE.1) WOU(K)=AU/(TOPL(K)-TOPL(K-1))
092               IS=J+1
093        12   CONTINUE
094     C
095           RETURN
096           END
```

```
001     C  SUBROUTINE: MET.MBARS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE UPPER AIR SOUNDING
005     C  HEIGHTS IN (MB) GIVEN HEIGHTS IN (M MSL).
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C       DP = ARRAY OF DEW POINT TEMP. OBS. IN SND. (DEG C)
009     C       NOBT = NUMBER OF OBSERVATIONS IN SOUNDING
010     C       THTM = ARRAY OF OBS. HEIGHTS IN SND. (M MSL)
011     C       TMP = ARRAY OF TEMP. OBS. IN SND. (DEG C)
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C       THTP = ARRAY OF OBS. HEIGHTS IN SND. (MB)
015     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
016     C  CONSTANT DEFINITIONS:
017     C       C = 0.034075  [(ACCEL. DUE TO GRAV)/UNIV. GAS CONST.)]
018     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
019     C  CALLING SUBROUTINES:
020     C       UPPER
021     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
022     C  CALLED SUBROUTINES:
023     C       (NONE)
024     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
025     C  COMMON BLOCKS:
026     C       (NONE)
027     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
028     C  PARAMETER DEFINITIONS:
029     C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
030     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
031     C
032           SUBROUTINE MBARS (NOBT,THTP,THTM,TMP,DP)
033     C
034           PARAMETER
035          +    NOB=30
036           DIMENSION
037          +    DP(NOB),       THTM(NOB),      THTP(NOB),      TMP(NOB)
038     C
039           C=0.034075
040           PP=0.
041     C
042     C              **SUCCESSIVE APPROXIMATION LOOP
043           DO 2 K=1,8
044             DO 1 I=2,NOBT
045               EL=6.11*10.**(7.5*DP(I-1)/(237.3+DP(I-1)))
046               EU=6.11*10.**(7.5*DP(I)/(237.3+DP(I)))
047               WL=622.*EL/(THTP(I-1)-EL)
048               IF (I.EQ.2)  THEN
049                 WU=WL
050               ELSE
051                 WU=622.*EU/(THTP(I)-EU)
052               ENDIF
053               TSTAR=((TMP(I)+TMP(I-1))/2.)+273.15+((WL+WU)/12.)
054               DELTAZ=THTM(I)-THTM(I-1)
055               DELTAP=-C*((THTP(I)+THTP(I-1))/2.)*DELTAZ/TSTAR
056               THTP(I)=THTP(I-1)+DELTAP
057           1   CONTINUE
058               IF (ABS(THTP(NOBT)-PP).LE..01) RETURN
059               PP=THTP(NOBT)
060           2 CONTINUE
061     C
062           RETURN
```

```
063          END
```

```
001     C  SUBROUTINE: MET.MIXSTB
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE GRIDDED MIXING HEIGHTS AND
005     C  P-G-T STABILITY CLASS FOR EACH PROCESSING TIME STEP.  INPUT DATA
006     C  IS USED FROM STATIONS HAVING BOTH UPPER AIR AND SRFC OBSERVATIONS.
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C  THE INPUTS ARE:
009     C    THRU ARGUMENT:
010     C      IJS = STARTING JULIAN DATE
011     C      IJE = ENDING JULIAN DATE
012     C      IHS = STARTING HOUR-OF-DAY (1 TO 24)
013     C      IHE = ENDING HOUR-OF-DAY (1 TO 24)
014     C      RAD = MAXIMUM SEPARATION DISTANCE, BETWEEN A SURFACE OR 'OTHER'
015     C            STATION AND AN UPPER AIR STATION, AT WHICH FOR ANY ACTUAL
016     C            SEPARATION DISTANCE GREATER THAN 'RAD', A 1/R**2
017     C            INTERPOLATION OF THE SOUNDING TO THE SURFACE OR 'OTHER'
018     C            STATION WILL NOT BE PERFORMED (KM).
019     C      NGAMA = NUMBER OF 'GAMMA' SURFACES
020     C      HTGAMA = ARRAY OF HEIGHT OF GAMMA SURFACES (GAMMA COORD)
021     C      NSTA = NUMBER OF UPPER AIR STATIONS
022     C      NSSTA = NUMBER OF SURFACE STATIONS
023     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
024     C                              8(PACIFIC)
025     C      TZNAM = BASE TIME ZONE NAME
026     C      DETAIL = LOGICAL VARIABLE IF .TRUE. THEN WRITE OUT INFORMATION
027     C               TO UNIT LUO
028     C      LUO = LOGICAL UNIT TO WHICH PRINTOUT IS DIRECTED
029     C    THRU COMMON BLOCK 'MAIN2':
030     C      IHH = 1  'SMALL H' = CONSTANT
031     C          = 2  'BIG H' = CONSTANT
032     C          = 3  'SMALL H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
033     C          = 4  'BIG H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
034     C      HH = VALUE FOR 'SMALL H' OR 'BIG H' DEPENDING ON WHETHER
035     C           IHH=1 OR IHH=2, RESPECTIVELY (M)
036     C      SLX = DOMAIN WIDTH IN X-DIR (KM)
037     C      SLY = DOMAIN WIDTH IN Y-DIR (KM)
038     C      RXY = SLX/SLY
039     C    THRU COMMON BLOCK 'UPPER1':
040     C      COX = ARRAY OF X-COORD OF UPPER AIR STATIONS (KM)
041     C      COY = ARRAY OF Y-COORD OF UPPER AIR STATIONS (KM)
042     C      IDU = ARRAY OF UPPER AIR STATIONS IDENTIFICATIONS
043     C      XLAT = ARRAY OF STATION LATITUDES (DEG)
044     C      YLONG = ARRAY OF STATION LONGITUDES (DEG)
045     C      IUSRTT = ARRAY OF STARTING RECORD NUMBER OF TEMP SNDGS FOR ALL
046     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 8
047     C      IUENDT = ARRAY OF ENDING RECORD NUMBER OF TEMP SNDGS FOR ALL
048     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 8
049     C      ITZU = ARRAY OF TIME ZONES WHERE STATIONS LOCATED
050     C    THRU COMMON BLOCK 'SRFC1':
051     C      IDS = ARRAY OF SURFACE STATIONS ID NUMBERS
052     C      ISSRT = ARRAY OF STARTING RECORD NUMBER OF SRFC OBS FOR ALL
053     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
054     C      ISEND = ARRAY OF ENDING RECORD NUMBER OF SRFC OBS FOR ALL
055     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
056     C    THRU LOGICAL UNIT 12:
057     C      FI = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR EACH 'GAMMA' SRFC
058     C      GI = ARRAY OF AMPLITUDE FUNCTION 'GSUGI' FOR EACH 'GAMMA' SRFC
059     C      GMF = ARRAY OF AMPLITUDE FUNCTION 'GSUBI-FSUBI' FOR EACH 'GAMMA'
060     C            SURFACE
061     C    THRU COMMON BLOCK 'GRID1':
062     C      IX0S = X-COORD ORIGIN OF MIXING HEIGHT AND STABILITY GRID (KM)
```

```
C       IYOS = Y-COORD ORIGIN OF MIXING HEIGHT AND STABILITY GRID (KM)
C       ICSS = CELL SIZE OF MIXING HEIGHT AND STABILITY GRID (KM)
C       NXCS = NUMBER OF GRID CELLS IN X-DIR
C       NYCS = NUMBER OF GRID CELLS IN Y-DIR
C       XR = ARRAY OF GRID POINTS X-COORDINATES (KM)
C       YR = ARRAY OF GRID POINTS Y-COORDINATES (KM)
C       A = ARRAY OF GRID POINTS ALPHA COORDINATES
C       B = ARRAY OF GRID POINTS BETA COORDINATES
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C    THRU COMMON BLOCK 'UPSFC1':
C      NMSTA = NUMBER OF STATIONS WITH BOTH SURFACE AND UPPER AIR OBS
C      IUM = ARRAY OF INDEX NUMBERS OF UPPER AIR STATIONS WITH SURFACE
C            OBSERVATIONS ALSO
C      ISM = ARRAY OF INDEX NUMBERS OF SURFACE STATIONS WITH UPPER AIR
C            OBSERVATIONS ALSO
C    THRU COMMON BLOCK 'AMPLFN':
C      FI = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR EACH 'GAMMA' SRFC
C      GI = ARRAY OF AMPLITUDE FUNCTION 'GSUGI' FOR EACH 'GAMMA' SRFC
C      GMF = ARRAY OF AMPLITUDE FUNCTION 'GSUBI-FSUBI' FOR EACH 'GAMMA'
C            SURFACE
C    THRU LOGICAL UNIT 17:
C      HMD = ARRAY OF GRIDDED MIXING HEIGHTS FOR TIME STEP
C      ISTAB = ARRAY OF GRIDDED STABILITY CLASSIFICATIONS FOR TIME STEP
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  FILES READ FROM:
C      LOGICAL UNIT 12 = UNFORMATTED SEQUENTIAL DISK FILE CONTAINING
C                        AMPLITUDE FUNCTIONS FOR EACH 'GAMMA' SRFC
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  FILES WRITTEN TO:
C      LOGICAL UNIT 17 = UNFORMATTED SEQUENTIAL DISK FILE CONTAINING
C                        GRIDDED MIXING HEIGHTS AND STABILITIES
C      LOGICAL UNIT LUO = IF DETAIL IS .TRUE. THEN RUN INFO IS OUTPUT
C      LOGICAL UNIT  6 = ERROR MESSAGES
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C      AAMAIN
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C      GRID, HPROF, SMALLH, SOLAR, STAB, TETRI, TPROF, WINDS, FINDU
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C      MAIN2, UPPER1, SRFC1, AMPLFN, UPSFC1, GRID1
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C      NGA = NUMBER OF 'GAMMA' SURFACES
C      NNS = NUMBER OF SURFACE STATIONS
C      NOB = NUMBER OF OBSERVATIONS IN A SOUNDING
C      NUA = NUMBER OF UPPER AIR STATIONS
C      NPO = NUMBER OF POLYNOMIAL TERMS
C      NPG = NUMBER OF GRID POINTS IN X-DIR OR Y-DIR FOR MIXING HEIGHT
C            AND STABILITY GRID
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE MIXSTB (IJS,IJE,IHS,IHE,NSTA,NSSTA,NGAMA,HTGAMA,RAD,
     +                   IBTZ,TZNAM,DETAIL,LUO)
C
      PARAMETER
     +    NGA=9,          NNS=15,         NOB=30,         NPO=21,
     +    NPG=10,         NUA=10
C
      LOGICAL
```

```
     +    DETAIL,          MORN,          AFT
C
      CHARACTER*10
     +    NAME,            NAMES,         TZNAM
C
      DIMENSION
     +    CMH(NUA,24),     HMD(NPG,NPG),  BLA(NOB),      ISTAB(NPG,NPG),
     +    PPT(NOB),        APT(24),       PTA(NOB),      PTM(NOB),
     +    TH(NOB),         THTA(NOB),     THTM(NOB),     XX(NUA),
     +    YY(NUA),         HTGAMA(NGA),   JMIN(NUA),     JCUT(NUA),
     +    IUPNT(NUA),      ISPNT(NNS)
C
      COMMON /MAIN2/
     +    HH,              IHH,           IX0,           IY0,
     +    RXY,             SLX,           SLY
C
      COMMON /AMPLFN/
     +    FI(NPO,NGA),     GI(NPO,NGA),   GMF(NPO,NGA)
C
      COMMON /UPPER1/
     +    COX(NUA),        COY(NUA),      EL(NUA),       IDU(NUA),
     +    NAME(NUA),       NSDT(NUA),     NSDW(NUA),     XLAT(NUA),
     +    YLONG(NUA),      IUSRTW(NUA),   IUSRTT(NUA),   IUENDW(NUA),
     +    IUENDT(NUA),     ITZU(NUA)
C
      COMMON /SRFC1/
     +    ELS(NNS),        IDS(NNS),      NAMES(NNS),    NOBS(NNS),
     +    SCOX(NNS),       SCOY(NNS),     SLAT(NNS),     SLONG(NNS),
     +    ISSRT(NNS),      ISEND(NNS)
C
      COMMON /UPSFC1/
     +    NMSTA,           IUM(NUA),      ISM(NNS)
C
      COMMON /GRID1/
     +    IX0S,            IY0S,          NXCS,          NYCS,
     +    ICSS,            A(NPG),        B(NPG),        XR(NPG),
     +    YR(NPG)
C
C              ***REWIND AMPLITUDE FILE.
      REWIND 12
      READ (12) IBL
      READ (12) IBL
C
C              ***WRITE DOMAIN SPECS TO UNIT 17.
      WRITE (17) IX0S,IY0S,NXCS,NYCS,ICSS
C
C              ***DETERMINE STATIONS TO BE USED IN COMPUTING MIXING
C              ***HEIGHT AND STABILITY.  THE STATIONS MUST HAVE UPPER
C              ***AIR AND SURFACE OBSERVATIONS.
      IC=0
      DO 3 J=1,NSTA
      DO 3 K=1,NSSTA
        IF (IDU(J).EQ.IDS(K))  THEN
          IC=IC+1
          IUM(IC)=J
          ISM(IC)=K
          IUPNT(IC)=IUSRTT(J)-1
          ISPNT(IC)=ISSRT(K)-1
        ENDIF
    3 CONTINUE
      NMSTA=IC
      IF (NMSTA.EQ.0)  THEN
```

```
187             WRITE (6,31)
188             STOP 'MIXSTB'
189           ENDIF
190     C
191     C*****************WRITE OUT RUN INFO TO UNIT LUO************************
192           WRITE(LUO,60) TZNAM
193           WRITE(LUO,61)
194           DO 32 L=1,NMSTA
195             WRITE(LUO,62)NAME(IUM(L)),IDU(IUM(L)),COX(IUM(L)),COY(IUM(L)),
196          +                XLAT(IUM(L)),YLONG(IUM(L)),EL(IUM(L))
197          32 CONTINUE
198           WRITE(LUO,68) IX0S,IY0S,NXCS,NYCS,ICSS
199     C*******************************************************************************
200     C
201     C               ***LOOP ON JULIAN DAY.
202           DO 100 JUL=IJS,IJE
203     C
204     C
205     C               ***LOOP ON STATIONS
206           LL=0
207           DO 23 L=1,NMSTA
208     C
209     C               ***COMPUTE TIME OF SUNRISE AND SUNSET
210           TSN=30.
211           CALL SOLAR(JUL,TSN,XLAT(IUM(L)),YLONG(IUM(L)),ITZU(IUM(L)),A1,
212          +           ALPHA,TSR,TSS)
213           ISR=NINT(TSR)+(ITZU(IUM(L))-IBTZ)
214           ISS=ISR+NINT(TSS-TSR)
215     C
216     C               *FIND MORNING SOUNDING.
217           MORN=.TRUE.
218           AFT=.FALSE.
219           CALL FINDU(JUL,ISR,ISS,IDU(IUM(L)),IUPNT(L),IUENDT(IUM(L)),MORN,
220          +           AFT,IHRM,HMM,NOBM,BLA,THTM,BLA,PTM)
221           IF(.NOT.MORN) GO TO 23
222     C
223     C               *FIND AFTERNOON SOUNDING
224           MORN=.FALSE.
225           AFT=.TRUE.
226           CALL FINDU(JUL,ISR,ISS,IDU(IUM(L)),IUPNT(L),IUENDT(IUM(L)),MORN,
227          +           AFT,IHRA,HMA,NOBA,BLA,THTA,BLA,PTA)
228           IF(.NOT.AFT) GO TO 23
229     C
230           DO 6 M=1,24
231             APT(M)=999.
232          6 CONTINUE
233     C
234     C               ***FIND SURFACE OBSERVATIONS FOR THIS STATION.
235          7 ISPNT(L)=ISPNT(L)+1
236             IF (ISPNT(L).GT.ISEND(ISM(L))) GO TO 23
237             READ (10'ISPNT(L)) IDD,IJUL,IHR,BL,BL,BL,PTT,BL,IBL
238             IF (IDD.NE.IDS(ISM(L))) THEN
239               WRITE (6,33)
240               STOP 'MIXSTB'
241             ENDIF
242             IF (IJUL-JUL) 7,43,21
243          43 IF (IHR.LT.ISR-4) GO TO 7
244             IF (IHR.GT.ISS+2) GO TO 23
245             APT(IHR)=PTT
246     C
247           8 ISPNT(L)=ISPNT(L)+1
248             IF (ISPNT(L).GT.ISEND(ISM(L))) GO TO 23
```

```
249           READ (10'ISPNT(L)) IDD,IJUL,IHR,BL,BL,BL,PTT,BL,IBL
250           IF (IDD.NE.IDS(ISM(L))) THEN
251             WRITE (6,33)
252             STOP 'MIXSTB'
253           ENDIF
254           IF (IJUL.GT.JUL)  THEN
255             ISPNT(L)=ISPNT(L)-1
256             GO TO 9
257           ENDIF
258           IF (IHR.GT.ISS+2) GO TO 9
259           APT(IHR)=PTT
260           GO TO 8
261         9 CONTINUE
262     C
263     C               ***CHECK TO SEE IF SURFACE POTENTIAL TEMPERATURE ARRAY
264     C               ***HAS AT LEAST 1 VALUE BEFORE SUNRISE, AT LEAST 1
265     C               ***VALUE AFTER SUNSET AND AT LEAST 1 VALUE DURING
266     C               ***MIDDAY (12 TO 15).  IF NOT GO TO NEXT STATION.
267           DO 11 J=ISR-4,ISR
268             IF (APT(J).NE.999.)  THEN
269               DO 10 K=ISS,ISS+2
270                 IF (APT(K).NE.999.) THEN
271                   DO 47 M=12,15
272                     IF (APT(M).NE.999.) GO TO 12
273        47           CONTINUE
274                     GO TO 21
275                   ENDIF
276        10       CONTINUE
277                 GO TO 21
278               ENDIF
279        11   CONTINUE
280             GO TO 21
281        12   CONTINUE
282     C
283           LL=LL+1
284     C
285     C               ***FILL SURFACE POTENTIAL TEMPERATURE ARRAY.
286     C
287     C               *SPECIFY VALUES AT ISR-4 AND ISS+2 IF VALUES MISSING
288           DO 14 J=ISR-4,ISR
289             IF (APT(J).NE.999.) THEN
290               DO 48 K=ISR-4,J-1
291                 APT(K)=APT(J)
292        48       CONTINUE
293                 GO TO 49
294               ENDIF
295        14   CONTINUE
296        49   CONTINUE
297             IF(APT(ISS+2).EQ.999.) THEN
298               IF(APT(ISS+1).EQ.999.) THEN
299                 APT(ISS+2)=APT(ISS)
300                 APT(ISS+1)=APT(ISS)
301               ELSE
302                 APT(ISS+2)=APT(ISS+1)
303               ENDIF
304             ENDIF
305     C
306     C               *LINEARLY INTERPOLATE TO GET MISSING VALUES.
307           IC=0
308           DO 16 J=ISR-4,ISS+2
309             IF (APT(J).NE.999.)  THEN
310               IF (IC.NE.0)  THEN
```

```
311                 DT=(APT(J)-APT(J-(IC+1)))/FLOAT(IC+1)
312                 DO 15 K=J-IC,J-1
313                   APT(K)=APT(K-1)+DT
314        15       CONTINUE
315                 IC=0
316               ENDIF
317             ELSE
318               IC=IC+1
319             ENDIF
320        16 CONTINUE
321     C
322     C               ***COMPUTE TEMPERATURE ADVECTION RATE.
323     C               ***COMPUTED AS THE POTENTIAL TEMPERATURE DIFFERENCE
324     C               ***BETWEEN THE MORNING AND AFTERNOON SOUNDINGS AT
325     C               ***3000 METERS ABOVE GROUND LEVEL.
326           HPMSL=3000.+THTM(1)
327           CALL TPROF (NOBM,THTM,PTM,HPMSL,VALM)
328           HPMSL=3000.+THTA(1)
329           CALL TPROF (NOBA,THTA,PTA,HPMSL,VALA)
330           TAR=(1./REAL(IHRA-IHRM))*(VALA-VALM)
331     C
332     C               ***FILL SURFACE POTENTIAL TEMPERATURE ARRAY FROM 4 HOURS
333     C               ***BEFORE SUNRISE TO 2 HOURS AFTER SUNSET WITH
334     C               ***RELATIVE TEMPERATURES.
335           IC=0
336           DO 17 J=ISR-4,ISS+2
337             APT(J)=APT(J)-REAL(IC)*TAR
338             IC=IC+1
339        17 CONTINUE
340     C
341     C               ***FIND MINIMUM RELATIVE TEMPERATURE FROM ISR-4
342     C               ***TO ISR+2
343           TMIN=APT(ISR-4)
344           JMIN(LL)=ISR-4
345           DO 50 J=ISR-3,ISR+2
346             IF(APT(J).LE.APT(J-1)) THEN
347               TMIN=APT(J)
348               JMIN(LL)=J
349             ENDIF
350        50 CONTINUE
351     C
352     C               ***FIND MAXIMUM AFTERNOON TEMPERATURE
353           TMAX=APT(ISR)
354           JMAX=ISR
355           DO 51 J=ISR+1,ISS+2
356             IF(APT(J).GE.APT(J-1)) THEN
357               TMAX=APT(J)
358               JMAX=J
359             ENDIF
360        51 CONTINUE
361     C
362     C               ***DETERMINE TEMPERATURE, AND HENCE TIME, AT WHICH
363     C               ***DAYTIME CONDITIONS END
364           TCUT=TMAX-0.25*(TMAX-TMIN)
365           JCUT(LL)=ISS+1
366           DO 52 J=JMAX,ISS+2
367             IF(APT(J).LE.TCUT) JCUT(LL)=J-1
368        52 CONTINUE
369     C
370     C               ***CORRECT MORNING SOUNDING TO SURFACE TEMPERATURE.
371           NOBT=NOBM
372           DO 18 J=1,NOBM
```

```
373             TH(J)=THTM(J)
374             PPT(J)=PTM(J)
375          18 CONTINUE
376             IF (PTM(1).NE.TMIN)  THEN
377               HMMSL=HMM+THTM(1)
378               IF (HMMSL.GE.THTM(NOBM))  THEN
379                 WRITE (6,34)
380                 STOP 'MIXSTB'
381               ENDIF
382               IF (HMM.LT.50.) HMM=50.
383               HMMSL=HMM+THTM(1)
384               CALL TPROF (NOBM,THTM,PTM,HMMSL,VALM)
385               PPT(1)=TMIN
386               PPT(2)=VALM
387               TH(1)=THTM(1)
388               TH(2)=HMMSL
389               IC=2
390               DO 19 J=2,NOBM
391                 IF (THTM(J).LE.HMMSL) GO TO 19
392                 IC=IC+1
393                 TH(IC)=THTM(J)
394                 PPT(IC)=PTM(J)
395          19   CONTINUE
396               NOBT=IC
397             ENDIF
398     C
399     C                ***COMPUTE CONVECTIVE MIXING DEPTHS FOR STATION.
400             DO 53 J=1,24
401               CMH(LL,J)=0.
402          53 CONTINUE
403             DO 20 J=JMIN(LL)+1,JCUT(LL)
404               IF (APT(J).LE.APT(J-1))  THEN
405                 CMH(LL,J)=CMH(LL,J-1)
406               ELSE
407                 CALL HPROF (NOBT,TH,PPT,APT(J),CMHMSL)
408                 CMH(LL,J)=CMHMSL-TH(1)
409               ENDIF
410          20 CONTINUE
411     C
412     C                ***SPECIFY STATION COORDINATES.
413             XX(LL)=COX(IUM(L))
414             YY(LL)=COY(IUM(L))
415             GO TO 23
416     C
417          21 CONTINUE
418             ISPNT(L)=ISPNT(L)-1
419     C
420     C                 ***END OF LOOP ON STATIONS
421          23 CONTINUE
422     C
423             NS=LL
424             IF (NS.EQ.0)  THEN
425               IF (JUL.EQ.IJS.AND.IHS.GT.ISR+2.OR.JUL.EQ.IJE.AND.IHE.LT.ISS-1)
426              +  THEN
427     C                **USE MECHANICAL MIXING HEIGHT ONLY.
428                 GO TO 25
429               ELSE
430                 WRITE (6,35) JUL
431                 STOP 'MIXSTB'
432               ENDIF
433             ENDIF
434     C
```

```
C             ***FIND MINIMUM JMIN AND MAXIMUM JCUT FOR ALL STATIONS
      JMINA=JMIN(1)
      JCUTA=JCUT(1)
      DO 54 J=2,NS
        IF(JMIN(J).LT.JMIN(J-1)) JMINA=JMIN(J)
        IF(JCUT(J).GT.JCUT(J-1)) JCUTA=JCUT(J)
   54 CONTINUE
C
C             ***STEP THROUGH EACH HOUR OF THE DAY COMPUTING MIXING
C             ***HEIGHT AND STABILITY FOR EACH GRID POINT.
   25 ISA=1
      ISP=24
      IF (JUL.EQ.IJS) ISA=IHS
      IF (JUL.EQ.IJE) ISP=IHE
      ICLKK=(JUL-1)*24
      DO 30 N=ISA,ISP
C***************WRITE OUT RUN INFO TO UNIT LUO************************
      IF(DETAIL) THEN
        WRITE(LUO,63) JUL,N
      ENDIF
C*******************************************************************************
      ICLK=ICLKK+N
      DO 26 J=1,NGAMA
        READ (12) ICK,THETTA,NTE,(FI(K,J),GI(K,J),GMP(K,J),K=1,NTE)
   26 CONTINUE
      IF (ICK.NE.ICLK)  THEN
        WRITE (6,39)
        STOP 'MIXSTB'
      ENDIF
C             ***COMPUTE CONSTANTS USED IN DENSITY CALCULATION
      ROA=353.1/THETTA
      HA=102.4*THETTA
C
C             ***LOOP ON GRID POINTS
      DO 29 J=1,NYCS
      DO 28 I=1,NXCS
C
C             ***COMPUTE TIME OF SUNRISE AND SUNSET AT GRID POINT.
      CALL GRID(XR(I),YR(J),XLATG,YLONGG)
      TSN=30.
      CALL SOLAR(JUL,TSN,XLATG,YLONGG,IBTZ,Al,ALPHA,TSR,TSS)
      ISR=NINT(TSR)
      ISS=ISR+NINT(TSS-TSR)
C
C             ***COMPUTE MECHANICAL MIXING DEPTH AT EACH GRID POINT.
      CALL TETRI (A(I),B(J),THH,DZA,DZB)
      CALL SMALLH (IHH,HH,THH,A(I),B(J),SH)
      IF (SH.GT.3000.) SH=3000.
      CALL WINDS (A(I),B(J),SH,NTE,NGAMA,HTGAMA,HA,ROA,HH,IHH,SLX,RXY,
     +            U,V,W,WT)
      SP=SQRT(U**2+V**2)
      HM=53.*SP
      IF (HM.LT.50.) HM=50.
C
      IF (NS.EQ.0)  THEN
        HMD(I,J)=HM
        CALL WINDS (A(I),B(J),10.,NTE,NGAMA,HTGAMA,HA,ROA,HH,IHH,SLX,
     +              RXY,U,V,W,WT)
        SP=SQRT(U**2+V**2)
        CALL STAB (ISR,ISS,N,HM,SP,ISTAB(I,J))
        GO TO 28
      ENDIF
```

```
497     C
498           IF (N.GT.JMINA.AND.N.LE.JCUTA)  THEN
499     C               **1/R**2 WEIGHT CONVECTIVE MIXING HEIGHTS TO GRID POINT.
500             WF=0.
501             HCMS=0.
502             DO 27 K=1,NS
503               RDIF=SQRT((XX(K)-XR(I))**2.+(YY(K)-YR(J))**2.)
504               IF (RDIF.LE.RAD)  THEN
505                 IF (RDIF.LE..1) RDIF=1.*10.**(-3.)
506                 WF=1./RDIF**2.+WF
507                 HCMS=(1/RDIF**2.)*CMH(K,N)+HCMS
508               ENDIF
509          27   CONTINUE
510             HCM=HCMS/WF
511             HMD(I,J)=AMAX1(HM,HCM)
512           ELSE
513             HMD(I,J)=HM
514           ENDIF
515     C               ***COMPUTE STABILITY.
516           CALL WINDS (A(I),B(J),10.,NTE,NGAMA,HTGAMA,HA,ROA,HH,IHH,SLX,RXY,
517          +            U,V,W,WT)
518           SP=SQRT(U**2+V**2)
519           CALL STAB (ISR,ISS,N,HCM,SP,ISTAB(I,J))
520          28 CONTINUE
521           WRITE (17) ICLK,(HMD(I,J),I=1,NXCS)
522           WRITE (17) ICLK,(ISTAB(I,J),I=1,NXCS)
523     C
524     C                     ***END OF GRID LOOP
525          29 CONTINUE
526     C****************WRITE OUT RUN INFO TO UNIT LUO************************
527           IF(DETAIL) THEN
528             WRITE(LUO,64) (XR(I),I=1,NXCS)
529             DO 70 J=NYCS,1,-1
530               WRITE(LUO,65) YR(J),(HMD(I,J),I=1,NXCS)
531          70   CONTINUE
532             WRITE(LUO,66) (XR(I),I=1,NXCS)
533             DO 71 J=NYCS,1,-1
534               WRITE(LUO,67) YR(J),(ISTAB(I,J),I=1,NXCS)
535          71   CONTINUE
536           ENDIF
537     C*******************************************************************************
538     C
539     C                     ***END OF HOUR LOOP
540          30 CONTINUE
541     C                     ***END OF DAY LOOP
542        100 CONTINUE
543     C
544           RETURN
545          31 FORMAT (2X,'NO STATIONS AVAILABLE FOR COMPUTING MIXING HEIGHTS',
546            +' AND STABILITIES.')
547          33 FORMAT (2X,'ID OF SURFACE DATA DOES NOT CORRESPOND WITH ID REQUIRE
548            +D.')
549          34 FORMAT (2X,'MECHANICAL MIXING HEIGHT GREATER THAN HIGHEST OBSERVAT
550            +ION.')
551          35 FORMAT (2X,'NO STATIONS FOR COMPUTATION OF MIXING HT. AND STB ON J
552            +ULIAN DAY',I5)
553          39 FORMAT (2X,'AMPLITUDE FILE CLOCK DOES NOT MATCH MIXSTB CLOCK.')
554          60 FORMAT(1H1,///,5X,'TRACE OF SUBROUTINE MIXSTB EXECUTION - MIXSTB',
555            +' COMPUTES HOURLY GRIDDED MIXING HEIGHTS AND STABILITIES FROM',
556            +/,5X,'UPPER AIR AND SURFACE TEMPERATURE OBSERVATIONS AND WINDS',
557            +' FROM THE FLOW MODEL.  THE BASE TIME ZONE IS',A10,'.' )
558          61 FORMAT(1H0,/,5X,'SUMMARY OF STATIONS USED:',/,10X,'NAME',
```

```
559           +6X,'ID NO',2X,'X (KM)',2X,'Y (KM)',2X,'LAT (DEG)',2X,
560           +'LONG (DEG)',2X,'ELEV (M MSL)')
561          62 FORMAT(1H ,6X,A10,2X,I6,2X,F6.1,2X,F6.1,3X,F7.3,5X,F7.3,6X,F6.1)
562          68 FORMAT(1H0,/,5X,'THE GRID SPECIFICATIONS ARE:',/,7X,'IX0S = ',I3,
563           +5X,'IY0S = ',I3,5X,'NXCS = ',I2,6X,'NYCS = ',I2,6X,'ICSS = ',I3)
564          63 FORMAT(1H0,/,5X,'JULIAN DAY ='1X,I3,5X,'HOUR ='1X,I2,10X,
565           +'(OUTPUT FROM SUBROUTINE MIXSTB)')
566          64 FORMAT(1H ,/,7X,'THE MIXING HEIGHTS ARE (M):',/,43X,'X (KM)',
567           +/,9X,'Y (KM)',1X,10(2X,F5.0),/)
568          65 FORMAT(1H ,9X,F4.0,2X,10(2X,F5.0))
569          66 FORMAT(1H ,/,7X,'THE STABILITY CLASSES ARE:',/,43X,'X (KM)',
570           +/,9X,'Y (KM)',1X,10(2X,F5.0),/)
571          67 FORMAT(1H ,9X,F4.0,2X,10(4X,I1,2X))
572             END
```

```
001     C  SUBROUTINE: MET.MONIN
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE MONIN-OBUKOV LENGTH
005     C  USING THE METHOD GIVEN IN MESOPUFF II (MARCH 1984).
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C       TSURF = AIR TEMPERATURE AT SURFACE (DEG K)
009     C       WT = W'THETA' ((M-DEGK)/S)
010     C       USTR = SURFACE SURFACE FRICTION VELOCITY (M/S)
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE OUTPUTS ARE:
013     C       CAPL = MONIN-OBUKOV LENGTH (M)
014     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
015     C  CONSTANT DEFINITIONS:
016     C       VK = 0.4  [VON KARMAN CONSTANT]
017     C       GRAV = 9.81  [ACCELERATION DUE TO GRAVITY (M/S**2)]
018     C       CONSTA = 1100.  [CONSTANT (S**2/M)]
019     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
020     C  CALLING SUBROUTINES:
021     C       BLPARS
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLED SUBROUTINES:
024     C       (NONE)
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  COMMON BLOCKS:
027     C       (NONE)
028     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
029     C
030           SUBROUTINE MONIN(TSURF,WT,USTR,CAPL)
031     C
032     C            ***CONSTANTS
033           VK=0.4
034           GRAV=9.81
035           CONSTA=1100.
036     C
037     C            ***CALCULATE MONIN-OBUKHOV LENGTH, CAPL, (METERS) FROM
038     C            ***USTAR AND W'THETA' (DAYTIME) OR USTAR (NIGHTTIME)
039     C
040           IF(WT.LE.0.0)GO TO 100
041           EL=-USTR**3*TSURF/(VK*GRAV*WT)
042           CAPL=AMAX1(EL,-9999.)
043           GO TO 500
044       100 CONTINUE
045           CAPL=CONSTA*USTR**2
046       500 CONTINUE
047     C
048           RETURN
049           END
```

```
001     C  SUBROUTINE: NEAR.MET
002     C  DATE: APRIL 1985
003     C
004     C    SUBROUTINE TO PLACE IN ORDER A SET OF POINTS NEAREST TO A
005     C  GIVEN POINT.  THE POINTS ARE ORDERED FROM NEAREST TO MOST DISTANT
006     C  FROM THE GIVEN POINT.  POINTS GREATER THAN DISTANCE 'RAD' FROM
007     C  GIVEN POINT ARE ASSIGNED A 999 DESIGNATOR.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       NSTA = NUMBER OF POINTS TO ORDER.
011     C       COX = ARRAY OF X-COORD. OF POINTS (GRAMA GRID KM)
012     C       COY = ARRAY OF Y-COORD. OF POINTS (GRAMA GRID KM)
013     C       SEGX = X-COORD. OF GIVEN POINT (GRAMA GRID KM)
014     C       SEGY = Y-COORD. OF GIVEN POINT (GRAMA GRID KM)
015     C       RAD = DISTANCE OF A POINT AWAY FROM GIVEN POINT AFTER WHICH
016     C             THE POINT WILL NOT BE RANKED BUT ASSIGNED A VALUE OF
017     C             999 (KM)
018     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
019     C  THE OUTPUTS ARE:
020     C       NE = ARRAY OF STATION RANKINGS (ELEMENT 1 REFERS TO RANKING
021     C            OF POINT 1 AND SO FORTH)
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLING SUBROUTINES:
024     C       COEFF
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  CALLED SUBROUTINES:
027     C       (NONE)
028     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
029     C  COMMON BLOCKS:
030     C       (NONE)
031     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
032     C  PARAMETER DEFINITIONS:
033     C       NUA = NUMBER OF POINTS TO BE RANKED (NUMBER OF UPPER AIR
034     C             STATIONS FOR THIS APPLICATION).
035     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
036     C
037             SUBROUTINE NEAR(NSTA,COX,COY,SEGX,SEGY,RAD,NE)
038     C
039             PARAMETER NUA=10
040     C
041             DIMENSION COX(NUA),COY(NUA),NE(NUA),DMIN(NUA)
042     C
043             DO 2 L=1,NSTA
044               DMIN(L)=999.
045           2 CONTINUE
046     C
047             DIST=SQRT((SEGX-COX(1))**2+(SEGY-COY(1))**2)
048             DMIN(1)=DIST
049             NE(1)=1
050             DO 10 L=2,NSTA
051               DIST=SQRT((SEGX-COX(L))**2+(SEGY-COY(L))**2)
052               DO 5 M=1,L-1
053                 IF(DIST.LT.DMIN(M)) THEN
054                   DO 4 J=L,M+1,-1
055                     DMIN(J)=DMIN(J-1)
056                     NE(J)=NE(J-1)
057           4         CONTINUE
058                     DMIN(M)=DIST
059                     NE(M)=L
060                     GO TO 6
061                   ENDIF
062           5     CONTINUE
```

```
063         6    IF(DMIN(L).EQ.999.) THEN
064                DMIN(L)=DIST
065                NE(L)=L
066              ENDIF
067        10 CONTINUE
068     C
069           DO 20 L=1,NSTA
070             IF(DMIN(L).GT.RAD) NE(L)=999
071        20 CONTINUE
072     C
073           RETURN
074           END
```

```
001     C  SUBROUTINE: MET.POLBAS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE POLYNOMIAL BASIS FUNCTIONS
005     C  'BI' AND 'CI' REFERENCED IN GRAMA REPORT 3 (AUG,1981) TABLE 3.
006     C  THESE ARE USED TO DEFINE THE MASS-CONSISTENT WIND FIELD.
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C  THE INPUTS ARE:
009     C       AL = ALPHA COORDINATE (TRANSFORMED X-COORD.)
010     C       BE = BETA COORDINATE (TRANSFORMED Y-COORD.)
011     C       NTERMS = NUMBER OF TERMS IN EXPANSION
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C       BI = ARRAY OF B SUB I TERMS.
015     C       CI = ARRAY OF C SUB I TERMS.
016     C       PBI = ARRAY OF PART. DER. OF BI WITH RESPECT TO ALPHA TERMS.
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  CONSTANT DEFINITIONS:
019     C       E = .01  [MINIMUM VALUE FOR ALPHA AND BETA]
020     C       NDEG = 15  [MAXIMUM DEGREE OF POLYNOMIAL.  THE MAXIMUM
021     C                NUM. OF TERMS IS 136 = (NDEG*(NDEG+1))/2+(NDEG+1)]
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLING SUBROUTINES:
024     C       AMPL, FROUDE, WINDS
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  CALLED SUBROUTINES:
027     C       (NONE)
028     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
029     C  COMMON BLOCKS:
030     C       (NONE)
031     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
032     C  PARAMETER DEFINITIONS:
033     C       NPO = NUMBER OF POLYNOMIAL TERMS DESCRIBING WIND COMPONENTS.
034     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
035     C
036           SUBROUTINE POLBAS (AL,BE,NTERMS,BI,CI,PBI)
037     C
038           PARAMETER
039          +    NPO=21
040     C
041           DIMENSION
042          +    BI(NPO),       CI(NPO),       PBI(NPO)
043     C
044     C             ***CONSTANTS
045           E=.01
046           NDEG=15
047     C
048     C              ***COMPUTE BASIS FUNCTIONS FOR ALL TERMS
049           ALPH=AL
050           BETA=BE
051           IF (ALPH.LT.E) ALPH=E
052           IF (BETA.LT.E) BETA=E
053           ICOUNT=0
054           DO 3 IPL=1,NDEG+1
055             DO 2 I=1,IPL
056               ICOUNT=ICOUNT+1
057               N=IPL-1
058               IC=N*(N+1)/2+I
059     C              **U-COMPONENTS
060               NP=N-I+1
061               NQ=I-1
062               BI(IC)=ALPH**NP*BETA**NQ
```

```
063     C            **W-COMPONENTS
064                NPP=N-I
065                NQP=I-1
066                NRP=N-I+1
067                PBI(IC)=NRP*ALPH**NPP*BETA**NQP
068     C            **V-COMPONENTS
069                NPB=N-I
070                NQB=I
071                IF (N.LT.1.OR.I.EQ.IPL) GO TO 1
072                CI(IC)=ALPH**NPB*BETA**NQB
073                AC1=NP
074                AC2=I
075                AC=-AC1/AC2
076                CI(IC)=AC*CI(IC)
077                IF (ICOUNT.EQ.NTERMS) RETURN
078           1    CONTINUE
079           2  CONTINUE
080     C            **LAST TERM IN V-COMPONENTS
081              N=IPL-1
082              IC=N*(N+3)/2+1
083              CI(IC)=ALPH**N
084              AC=-1.
085              CI(IC)=AC*CI(IC)
086              IF (ICOUNT.EQ.NTERMS) RETURN
087           3 CONTINUE
088     C
089            RETURN
090            END
```

```
001     C  SUBROUTINE: MET.RDM
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO PROVIDE A RANDOM ADJUSTMENT TO
005     C  WIND DIRECTION OBSERVATIONS WHERE THE OBSERVATIONS ARE REPORTED
006     C  TO THE NEAREST 10 DEGREES.  THE ADJUSTMENTS ARE BETWEEN -4 TO
007     C  5 DEGREES.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       DS = THE SEED NUMBER
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE OUTPUTS ARE:
013     C       OUT = THE ADJUSTMENT BETWEEN -4 TO 5 DEGREES.
014     C       DS = THE NEXT SEED NUMBER
015     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
016     C  CALLING SUBROUTINES:
017     C       SRFC, UPPER
018     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
019     C  CALLED SUBROUTINES:
020     C       (NONE)
021     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
022     C  COMMON BLOCKS:
023     C       (NONE)
024     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
025     C
026             SUBROUTINE RDM (DS,OUT)
027     C
028             DOUBLE PRECISION
029            +    D1,              DM,              DS
030             DIMENSION
031            +    COR(10)
032     C
033             DATA DM/2147483647.D0/
034             DATA D1/2147483648.D0/
035             DATA COR/-4.,-3.,-2.,-1.,0.,1.,2.,3.,4.,5./
036     C
037             DS=DMOD(16807.D0*DS,DM)
038             OUTT=DS/D1
039             NN=INT(OUTT*10.)+1
040             IF (NN.GT.10) NN=10
041             IF (NN.LE.0) NN=1
042             OUT=COR(NN)
043     C
044             RETURN
045             END
```

```
001     C  SUBROUTINE: MET.READS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO PROVIDE TWO SURFACE OBS. RECORDS
005     C  FROM THE DIRECT ACCESS DISK FILE ON UNIT 10 TO COMMON BLOCK
006     C  'REINTS'.  THE TIME OF ONE RECORD IS LESS THAN OR EQUAL TO THE
007     C  CURRENT PROCESS TIME (ICLK), AND THE TIME OF THE SECOND RECORD
008     C  IS GREATER THEN ICLK.  THIS SUBROUTINE ALSO COMPUTES THE
009     C  OBSERVATIONS AT TIME ICLK BY LINEARLY INTERPOLATING BETWEEN THE
010     C  OBSERVATIONS AT TIME ICLKS1 AND ICLKS2.  IF THE TWO OBSERVATIONS
011     C  ARE GREATER THAN 8 HOURS APART, THE INTERPOLATED OBSERVATIONS
012     C  WILL NOT BE COMPUTED AND THE STATION WILL BE SET INACTIVE.
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  THE INPUTS ARE:
015     C    THRU ARGUMENT:
016     C      ICLK = CURRENT TIME BEING PROCCESSED (HOUR OF YEAR)
017     C           = (JULIAN DAY - 1)*24 + (HOUR OF DAY)
018     C      NOBSS = NUMBER OF RECORDS ALREADY READ FOR STATION K
019     C      ISTREC = STARTING RECORD ON DIRECT ACCESS DISK FILE OF
020     C               STATION K
021     C      NOBS = TOTAL NUMBER OF RECORDS FOR STATION K
022     C      K = ARRAY INDEX NUMBER OF STATION BEING PROCCESSES (1 THRU
023     C          TOTAL NUMBER OF STATIONS BEING PROCCESSED)
024     C      RDS1ST = LOGICAL VARIABLE IF .TRUE. MEANS FIRST CALL TO
025     C               SUBROUTINE FOR STATION K, IN WHICH CASE, AT LEAST
026     C               THE FIRST TWO RECORDS ARE READ
027     C    THRU LOGICAL UNIT 10:
028     C      IDD = STATION ID OF RECORD BEING READ
029     C      IJULS2 = JULIAN DAY OF RECORD BEING READ
030     C      IHRS2 = HOUR OF DAY OF RECORD BEING READ (1 TO 24 MST)
031     C      WXS2 = X-COMP. OF WIND OBSERVATION FOR STATION K
032     C      WYS2 = Y-COMP. OF WIND OBSERVATION FOR STATION K
033     C      TSURF2 = TEMPERATURE AT STATION K
034     C      PT = POTENTIAL TEMPERATURE AT STATION K
035     C      RHO2 = AIR DENSITY AT STATION K
036     C      ICC2 = CLOUD COVER AT STATION K
037     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
038     C  THE OUTPUTS ARE:
039     C    THRU ARGUMENT:
040     C      NOBSS = NUMBER OF RECORDS READ FOR STATION K
041     C      PASS = LOGICAL VARIABLE IF .TRUE. MEANS NO INTERPOLATED DATA
042     C             AVAILABLE AT TIME ICLK FOR STATION K
043     C      WXS = INTERPOLATED X-COMP. OF WIND OBS. FOR STATION K AT ICLK
044     C      WYS = INTERPOLATED Y-COMP. OF WIND OBS. FOR STATION K AT ICLK
045     C      TSURF = INTERPOLATED TEMPERATURE FOR STATION K AT TIME ICLK
046     C      RHO = INTERPOLATED AIR DENSITY FOR STATION K AT TIME ICLK
047     C      ICC = INTERPOLATED CLOUD COVER FOR STATION K AT TIME ICLK
048     C    THRU COMMON BLOCK REINTS:
049     C      ICLKS1 = HOUR-OF-YEAR OF PREVIOUS RECORD READ FOR STATION
050     C               K; ICLKS1 IS LESS THAN ICLK
051     C      ICLKS2 = HOUR-OF-YEAR OF CURRENT RECORD READ FOR STATION
052     C               K; ICLKS2 IS GREATER THAN ICLK
053     C      ICLKSD = ICLKS2 - ICLKS1
054     C      WXS1 = X-COMP. OF WIND OBSERVATION FOR STATION K AT ICLKS1
055     C      WYS1 = Y-COMP. OF WIND OBSERVATION FOR STATION K AT ICLKS1
056     C      TSURF1 = TEMPERATURE FOR STATION K AT TIME ICLKS1
057     C      RHO1 = AIR DENSITY FOR STATION K AT TIME ICLKS1
058     C      ICC1 = CLOUD COVER FOR STATION K AT TIME ICLKS1
059     C      WXS2 = SAME AS WXS1 EXCEPT FOR TIME ICLKS2
060     C      WYS2 = SAME AS WYS1 EXCEPT FOR TIME ICLKS2
061     C      TSURF2 = SAME AS TSURF1 EXCEPT FOR TIME ICLKS2
062     C      RHO2 = SAME AS RHO1 EXCEPT FOR TIME ICLKS2
```

```
063     C       ICC2 = SAME AS ICC1 EXCEPT FOR TIME ICLKS2
064     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
065     C  CONSTANTS DEFINITIONS:
066     C       INTSPN = 8   [MAX. TIME BETWEEN TWO OBSERVATIONS IN WHICH
067     C                     VALUES WILL STILL BE INTERPOLATED.]
068     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
069     C  FILES READ FROM:
070     C       LOGICAL UNIT 10 = DIRECT ACCESS DISK FILE, OF SURFACE WEATHER
071     C                         OBSERVATIONS CREATED BY SUBROUTINE 'SRFC'
072     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
073     C  CALLING SUBROUTINES:
074     C       AMPLIT, BLPARS
075     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
076     C  CALLED SUBROUTINES:
077     C       (NONE)
078     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
079     C  COMMON BLOCKS:
080     C       REINTS  [USED TO STORE TWO OBSERVATION RECORDS]
081     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
082     C  PARAMETER DEFINITIONS:
083     C       NNS = NUMBER OF SURFACE WEATHER STATIONS
084     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
085     C
086           SUBROUTINE READS (ICLK,NOBSS,ISTREC,NOBS,K,RDS1ST,PASS,WXS,
087          +                  WYS,TSURF,RHO,ICC)
088     C
089           PARAMETER
090          +    NNS=15
091     C
092           LOGICAL
093          +    PASS,          RDS1ST
094     C
095           COMMON /REINTS/
096          +    ICLKS1(NNS),   ICLKS2(NNS),   ICLKSD(NNS),   WXS1(NNS),
097          +    WXS2(NNS),     WYS1(NNS),     WYS2(NNS),     TSURF1(NNS),
098          +    TSURF2(NNS),   RHO1(NNS),     RHO2(NNS),     ICC1(NNS),
099          +    ICC2(NNS)
100     C
101           INTSPN=8
102           PASS=.FALSE.
103     C
104     C            ***IF FIRST CALL TO SUBROUTINE THEN INITIALIZE THE
105     C            ***FOLLOWING PARAMETERS TO ALLOW THE READING OF
106     C            ***AT LEAST THE FIRST TWO RECORDS FOR STATION K.
107           IF(RDS1ST) THEN
108             NOBSS=0
109             ICLKS2(K)=0
110           ENDIF
111     C
112           IF(ICLKS2(K).LT.ICLK) THEN
113     C            ***UPDATE TWO SURFACE OBSERVATION RECORDS.
114     C            **CHECK NUMBER OF OBS. READ FROM STATION K.
115         3   NOBSS=NOBSS+1
116             IF (NOBSS.GT.NOBS) THEN
117               PASS=.TRUE.
118               RETURN
119             ENDIF
120     C            **REASSIGN '2' TO '1'.
121             WXS1(K)=WXS2(K)
122             WYS1(K)=WYS2(K)
123             TSURF1(K)=TSURF2(K)
124             RHO1(K)=RHO2(K)
```

```
125             ICC1(K)=ICC2(K)
126             ICLKS1(K)=ICLKS2(K)
127     C               **READ IN NEXT SURFACE OBSERVATION.
128             READ (10'ISTREC+NOBSS-1) IDD,IJULS2,IHRS2,WXS2(K),WYS2(K),
129          +                           TSURF2(K),PT,RHO2(K),ICC2(K)
130     C               **DETERMINE CLOCK TIME OF OBSERVATION.
131             ICLKS2(K)=(IJULS2-1)*24+IHRS2
132             ICLKSD(K)=ICLKS2(K)-ICLKS1(K)
133             IF(ICLKS1(K).LE.0) GO TO 3
134             IF(ICLKS2(K).LT.ICLK) GO TO 3
135           ENDIF
136     C
137     C               ***INTERPOLATE BETWEEN TWO OBSERVATIONS TO GET VALUES AT
138     C               ***DESIRED TIME ICLK.
139           IF(ICLKS1(K).LE.ICLK) THEN
140             IF(ICLKSD(K).LE.INTSPN) THEN
141     C               **COMPUTE THE WEIGHTING FACTORS.
142               DEN=REAL(ICLKS2(K))-REAL(ICLKS1(K))
143               WF2=(REAL(ICLK)-REAL(ICLKS1(K)))/DEN
144               WF1=(REAL(ICLKS2(K))-REAL(ICLK))/DEN
145     C               **COMPUTE OBSERVATIONS AT ACTUAL TIME.
146               WXS=WF1*WXS1(K)+WF2*WXS2(K)
147               WYS=WF1*WYS1(K)+WF2*WYS2(K)
148               TSURF=WF1*TSURF1(K)+WF2*TSURF2(K)
149               RHO=WF1*RHO1(K)+WF2*RHO2(K)
150               ICC=NINT(WF1*FLOAT(ICC1(K))+WF2*FLOAT(ICC2(K)))
151               IF(ICC.GT.10) ICC=10
152               IF(ICC.LT.0) ICC=0
153             ELSE
154               PASS=.TRUE.
155             ENDIF
156           ELSE
157             PASS=.TRUE.
158           ENDIF
159     C
160           RETURN
161           END
```

```
001     C  SUBROUTINE: MET.READU
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO PROVIDE TWO UPPER AIR WIND SOUNDINGS
005     C  FROM THE DIRECT ACCESS DISK FILE ON UNIT 8 TO COMMON BLOCK
006     C  'REINTU'.  THE TIME OF ONE SOUNDING IS LESS THAN OR EQUAL TO THE
007     C  CURRENT PROCESS TIME (ICLK), AND THE TIME OF THE SECOND SOUNDING
008     C  IS GREATER THEN ICLK.  THIS SUBROUTINE ALSO COMPUTES THE WINDS AT
009     C  TIME 'ICLK' BY LINEARLY INTERPOLATING THE WINDS BETWEEN THE
010     C  SOUNDING AT TIME ICLK1 AND THE SOUNDING AT TIME ICLK2.  IF THE
011     C  TWO SOUNDINGS ARE GREATER THAN 24 HOURS APART, THE INTERPOLATED
012     C  WINDS WILL NOT BE COMPUTED AND THE STATION WILL BE SET INACTIVE.
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  THE INPUTS ARE:
015     C    THRU ARGUMENT:
016     C      ICLK = CURRENT TIME BEING PROCCESSED (HOUR OF YEAR)
017     C           = (JULIAN DAY - 1)*24 + (HOUR OF DAY)
018     C      NSND = NUMBER OF SOUNDINGS ALREADY READ FOR STATION JJ
019     C      ISTREC = STARTING RECORD ON DIRECT ACCESS DISK FILE OF
020     C               STATION JJ
021     C      NSDW = TOTAL NUMBER OF SOUNDINGS FOR STATION JJ
022     C      NGAMA = NUMBER OF 'GAMMA' SURFACES THAT WINDS ARE AVAIL.
023     C      JJ = ARRAY INDEX NUMBER OF STATION BEING PROCCESSES (1 THRU
024     C         TOTAL NUMBER OF STATIONS BEING PROCCESSED)
025     C      RDU1ST = LOGICAL VARIABLE IF .TRUE. MEANS FIRST CALL TO
026     C               SUBROUTINE FOR STATION JJ, IN WHICH CASE, AT LEAST
027     C               THE FIRST TWO SOUNDINGS ARE READ
028     C    THRU LOGICAL UNIT 8:
029     C      IID2 = STATION ID OF SOUNDING BEING READ
030     C      IJUL2 = JULIAN DAY OF SOUNDING BEING READ
031     C      IHR2 = HOUR OF DAY OF SOUNDING BEING READ (1 TO 24 MST)
032     C      WX2 = ARRAY OF X-COMP. OF WINDS ON ALL 'GAMMA' SURFACES
033     C            FOR STATION JJ
034     C      WY2 = ARRAY OF Y-COMP. OF WINDS ON ALL 'GAMMA' SURFACES
035     C            FOR STATION JJ
036     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
037     C  THE OUTPUTS ARE:
038     C    THRU ARGUMENT:
039     C      NSND = NUMBER OF SOUNDINGS READ FOR STATION JJ
040     C      PASS = LOGICAL VARIABLE IF .TRUE. MEANS NO INTERPOLATED WIND
041     C             DATA AVAILABLE AT TIME ICLK FOR STATION JJ
042     C      WX = ARRAY OF INTERPOLATED X-COMP. WIND FOR ALL 'GAMMA'
043     C           SURFACES FOR STATION JJ AT TIME ICLK
044     C      WY = ARRAY OF INTERPOLATED Y-COMP. WIND FOR ALL 'GAMMA'
045     C           SURFACES FOR STATION JJ AT TIME ICLK
046     C    THRU COMMON BLOCK REINTU:
047     C      ICLK1 = HOUR-OF-YEAR OF PREVIOUS SOUNDING READ FOR STATION
048     C              JJ; ICLK1 IS LESS THAN ICLK
049     C      ICLK2 = HOUR-OF-YEAR OF CURRENT SOUNDING READ FOR STATION
050     C              JJ; ICLK2 IS GREATER THAN ICLK
051     C      ICLKD = ICLK2 - ICLK1
052     C      WX1 = ARRAY OF X-COMP. WIND FOR ALL 'GAMMA' SURFACES FOR
053     C            STATION JJ AT TIME ICLK1
054     C      WY1 = ARRAY OF Y-COMP. WIND FOR ALL 'GAMMA' SURFACES FOR
055     C            STATION JJ AT TIME ICLK1
056     C      WX2 = SAME AS WX1 EXCEPT FOR TIME ICLK2
057     C      WY2 = SAME AS WY1 EXCEPT FOR TIME ICLK2
058     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
059     C  CONSTANTS DEFINITIONS:
060     C      INTSPX = 24 HRS  [MAX. TIME BETWEEN TWO SOUNDINGS IN WHICH
061     C                        VALUES WILL BE STILL BE INTERPOLATED]
062     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
063     C  FILES READ FROM:
064     C       LOGICAL UNIT 8 = DIRECT ACCESS DISK FILE, OF UPPER AIR WIND
065     C                        DATA, CREATED BY SUBROUTINE 'UPPER'
066     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
067     C  CALLING SUBROUTINES:
068     C       AMPLIT
069     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
070     C  CALLED SUBROUTINES:
071     C       (NONE)
072     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
073     C  COMMON BLOCKS:
074     C       REINTU  [USED TO STORE TWO SOUNDINGS]
075     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
076     C  PARAMETER DEFINITIONS:
077     C       NGA = NUMBER OF 'GAMMA' SURFACES WITH WIND DATA
078     C       NUA = NUMBER OF UPPER AIR STATIONS
079     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
080     C
081           SUBROUTINE READU (ICLK,NSND,ISTREC,NSDW,NGAMA,JJ,RDU1ST,
082          +                  PASS,WX,WY)
083     C
084           PARAMETER
085          +    NGA=9,          NUA=10
086     C
087           LOGICAL
088          +    PASS,           RDU1ST
089     C
090           DIMENSION
091          +    WX(NGA),        WY(NGA)
092     C
093           COMMON /REINTU/
094          +    ICLK1(NUA),     ICLK2(NUA),     ICLKD(NUA),     WX1(NUA,NGA),
095          +    WX2(NUA,NGA),   WY1(NUA,NGA),   WY2(NUA,NGA)
096     C
097           INTSPN=24
098           PASS=.FALSE.
099     C
100     C             ***IF FIRST CALL TO SUBROUTINE FOR STATION JJ THEN
101     C             ***INITIALIZE FOLLOWING PARAMETERS TO ALLOW READING
102     C             ***IN OF AT LEAST THE FIRST TWO SOUNDINGS.
103           IF(RDU1ST) THEN
104             NSND=0
105             ICLK2(JJ)=0
106           ENDIF
107     C
108           IF(ICLK2(JJ).LT.ICLK) THEN
109     C             ***UPDATE TWO SOUNDINGS
110     C             **CHECK NUMBER OF SOUNDINGS READ FROM STATION JJ.
111         4   NSND=NSND+1
112             IF (NSND.GT.NSDW) THEN
113               PASS=.TRUE.
114               RETURN
115             ENDIF
116     C             **REASSIGN '2' TO '1'.
117             DO 1 J=1,NGAMA
118               WX1(JJ,J)=WX2(JJ,J)
119               WY1(JJ,J)=WY2(JJ,J)
120         1   CONTINUE
121             ICLK1(JJ)=ICLK2(JJ)
122     C             **READ IN NEXT SOUNDING.
123             READ (8'ISTREC+NSND-1) IID2,IJUL2,IHR2,(WX2(JJ,K),WY2(JJ,K),K=1
124          +  ,NGAMA)
```

```
125     C             **DETERMINE CLOCK TIME OF SOUNDING.
126             ICLK2(JJ)=(IJUL2-1)*24+IHR2
127             ICLKD(JJ)=ICLK2(JJ)-ICLK1(JJ)
128             IF(ICLK1(JJ).LE.0) GO TO 4
129             IF(ICLK2(JJ).LT.ICLK) GO TO 4
130           ENDIF
131     C
132     C               ***INTERPOLATE BETWEEN TWO SOUNDINGS TO GET VALUES AT
133     C               ***DESIRED TIME ICLK.
134           IF(ICLK1(JJ).LE.ICLK) THEN
135             IF(ICLKD(JJ).LE.INTSPN) THEN
136     C                 **COMPUTE WEIGHTING FACTORS.
137               DEN=REAL(ICLK2(JJ))-REAL(ICLK1(JJ))
138               WF2=(REAL(ICLK)-REAL(ICLK1(JJ)))/DEN
139               WF1=(REAL(ICLK2(JJ))-REAL(ICLK))/DEN
140     C               **COMPUTE WINDS AT ICLK.
141               DO 7 J=1,NGAMA
142                 WX(J)=WF1*WX1(JJ,J)+WF2*WX2(JJ,J)
143                 WY(J)=WF1*WY1(JJ,J)+WF2*WY2(JJ,J)
144           7     CONTINUE
145             ELSE
146               PASS=.TRUE.
147             ENDIF
148           ELSE
149             PASS=.TRUE.
150           ENDIF
151     C
152           RETURN
153           END
```

```
001     C  SUBROUTINE: MET.SHSLP
002     C  DATE: APRIL 1985
003     C
004     C     SUBROUTINE TO EVALUATE THE PARTIAL DERIVITIVE OF 'SMALL H'
005     C  WITH RESPECT TO ALPHA (DHA) AND WITH RESPECT TO BETA (DHB).
006     C  THE RESULTS ARE USED IN THE COMPUTATION OF THE MASS-CONSISTENT
007     C  WINDS.  'SMALL H' IS THE DISTANCE BETWEEN THE TERRAIN SURFACE
008     C  AND THE UPPER BOUNDARY TO THE FLOW.
009     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
010     C  THE INPUTS ARE:
011     C       IHH = OPTION NUMBER IDENTIFYING FUNCTIONAL FORM OF 'SMALL H'.
012     C             (REFER TO SUBROUTINE SMALLH FOR EXPLAINATION)
013     C       DZA = TERRAIN SLOPE WITH RESPECT TO ALPHA.
014     C       DZB = TERRAIN SLOPE WITH RESPECT TO BETA.
015     C       A = ALPHA (TRANSFORMED X-COORDINATE).
016     C       B = BETA (TRANSFORMED Y-COORDINATE).
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       DHA = SLOPE OF SMALL H WITH RESPECT TO ALPHA.
020     C       DHB = SLOPE OF SMALL H WITH RESPECT TO BETA.
021     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
022     C  CALLING SUBROUTINES:
023     C       FROUDE, WINDS
024     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
025     C  CALLED SUBROUTINES:
026     C       (NONE)
027     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
028     C  COMMON BLOCKS:
029     C       (NONE)
030     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
031     C
032           SUBROUTINE SHSLP (IHH,DZA,DZB,A,B,DHA,DHB)
033     C
034           GO TO (1,2,3,4), IHH
035     C
036     C                **SMALL H = CONSTANT.
037         1 DHA=0.
038           DHB=0.
039           GO TO 5
040     C
041     C                **BIG H = CONSTANT.
042         2 DHA=-DZA
043           DHB=-DZB
044           GO TO 5
045     C
046     C                **SMALL H SOME FUNCTIONAL FORM OF A AND B.
047     C                **ANALYTICAL FORM ON P89 OF GRAMA 3 IN NOW.
048         3 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 1
049           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 1
050           DHA=100.*EXP(A-B)
051           DHB=-100.*EXP(A-B)
052           GO TO 5
053     C
054     C                **BIG H IS SOME FUNCTIONAL FORM OF A AND B.
055     C                **DHA=BIGH,A(A,B) - DZA
056     C                **DHB=BIGH,B(A,B) - DZB
057         4 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 2
058           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 2
059           GO TO 5
060     C
061         5 RETURN
062           END
```

```
001     C  PROGRAM: MET.SMALLH
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO DETERMINE 'SMALL H' WHERE 'SMALL H'
005     C  IS THE DISTANCE BETWEEN THE TERRAIN SURFACE AND THE UPPER
006     C  BOUNDARY TO THE FLOW.  'SMALL H' = 'BIG H' - 'TERRAIN HGHT',
007     C  WHERE 'BIG H' IS THE HEIGHT (M MSL) OF THE UPPER BOUNDARY.
008     C  'SMALL H' IS USED IN THE COMPUTATION OF THE MASS-CONSISTENT
009     C  WINDS.
010     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
011     C  THE INPUTS ARE:
012     C       IHH = 1   'SMALL H' = CONSTANT
013     C           = 2   'BIG H' = CONSTANT
014     C           = 3   'SMALL H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
015     C           = 4   'BIG H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
016     C       HH = VALUE FOR 'SMALL H' OR 'BIG H' DEPENDING ON WHETHER
017     C            IHH=1 OR IHH=2, RESPECTIVELY. (M) [IF STATION OUTSIDE
018     C            DOMAIN FOR IHH=3 OR 4, THEN 'SMALL H' OR 'BIG H' IS
019     C            SET EQUAL TO HH.]
020     C       TH = TERRAIN HEIGHT (M MSL)
021     C       A = ALPHA (TRANSFORMED X-COORDINATE)
022     C       B = BETA (TRANSFORMED Y-COORDINATE)
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  THE OUTPUTS ARE:
025     C       SH = 'SMALL H' (M)
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLING SUBROUTINES:
028     C       AMPL, FROUDE, MIXSTB, UPPER, WADJ, WINDS
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  CALLED SUBROUTINES:
031     C       (NONE)
032     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
033     C  COMMON BLOCKS:
034     C       (NONE)
035     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
036     C
037           SUBROUTINE SMALLH (IHH,HH,TH,A,B,SH)
038     C
039           GO TO (1,2,3,4), IHH
040     C
041     C                **SMALL H = CONSTANT.
042         1 SH=HH
043           GO TO 5
044     C
045     C                **BIG H = CONSTANT.
046         2 SH=HH-TH
047           GO TO 5
048     C
049     C                **FUNCTIONAL FORM OF SMALL H.
050     C                **THE ANALYTICAL FUNCTION ON P89 OF GRAMA 3 IS IN NOW.
051         3 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 1
052           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 1
053           SH=2000.+100.*EXP(A-B)
054           GO TO 5
055     C
056     C                **FUNCTIONAL FORM OF BIG H.
057     C                **MUST INSERT H(A,B) WHERE SH=H(A,B)-TH.
058         4 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 2
059           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 2
060           GO TO 5
061     C
062         5 RETURN
```

```
063          END
```

```
001     C   SUBROUTINE: MET.SOLAR
002     C   DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO CALCULATE: TIME OF SUNRISE AND TIME OF
005     C  SUNSET FOR ANY DAY, AND THE SIN(SOLAR ELEVATION ANGLE) FOR ANY
006     C  DAY FOR ANY HOUR; OR THE TIME OF SUNRISE, TIME OF SUNSET, AND THE
007     C  SOLAR FLUX AT SOLAR NOON FOR ANY DAY.  THE TIME IS IN STANDARD TIME
008     C  FOR THE CHOSEN TIME ZONE.  THIS SUBROUTINE IS A COMBINATION OF
009     C  SUBROUTINE SOLAR FROM MESOPAC II AND SUBROUTINE SOLAR FROM VALMET.
010     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
011     C  THE INPUTS ARE:
012     C      JULDY = JULIAN DAY OF THE YEAR (0 TO 366)
013     C      HR = HOUR OF THE DAY (STANDARD TIME).
014     C           ALSO ACTS AS A FLAG; IF HR=30. THEN SUBROUTINE
015     C           COMPUTES A1, PLUS ALPHA AT SOLAR NOON, HR OF SOLAR
016     C           NOON, AND TSR AND TSS.
017     C      SLAT = LATITUDE OF DESIRED LOCATION (DEG)
018     C      SLONG = LONGITUDE OF DESIRED LOCATION (DEG)
019     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
020     C                              8(PACIFIC)
021     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
022     C  THE OUTPUTS ARE:
023     C      A1 = SOLAR FLUX AT SOLAR NOON (WATTS/M**2)
024     C      ALPHA = SIN(SOLAR ELEVATION ANGLE)
025     C      TSR = TIME OF SUNRISE (STANDARD TIME IN CHOSEN TIME ZONE)
026     C      TSS = TIME OF SUNSET (STANDARD TIME IN CHOSEN TIME ZONE)
027     C      HR = TIME OF SOLAR NOON WHEN HR INPUT AS 30.
028     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
029     C  CALLING SUBROUTINES:
030     C      BLPARS, COEFF, MIXSTB, TEMP
031     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
032     C  CALLED SUBROUTINES:
033     C      (NONE)
034     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
035     C  COMMON BLOCKS:
036     C      (NONE)
037     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
038     C
039           SUBROUTINE SOLAR(JULDY,HR,SLAT,SLONG,IBTZ,A1,ALPHA,TSR,TSS)
040     C
041     C     CONSTANT 0.9856479 IS 360./365.242
042           D=(FLOAT(JULDY)-1.)*0.9856479
043           RADD=0.0174533*D
044           SIND=SIN(RADD)
045           COSD=COS(RADD)
046           RAD2D=2.*RADD
047           SIN2D=SIN(RAD2D)
048           COS2D=COS(RAD2D)
049           EM=12.+0.12357*SIND-0.004289*COSD+0.153809*SIN2D+0.060783*COS2D
050           SIGMA=279.9348+D+1.914827*SIND-0.079525*COSD+0.019938*SIN2D
051          1-0.00162*COS2D
052           SINCD=0.39784989*SIN(0.0174533*SIGMA)
053           CAPD=ASIN(SINCD)
054           COSCD=COS(CAPD)
055           RADLAT=0.0174533*SLAT
056           SINLAT=SIN(RADLAT)
057           COSLAT=COS(RADLAT)
058     C     **COMPUTE TIME SUNRISE AND SUNSET
059     C     CONSTANT 3.8197187 IS 57.29578/15.
060           CAPH=ACOS(-SINLAT*SINCD/(COSLAT*COSCD))*3.8197187
061           DTEMP=SLONG/15.+EM-IBTZ
062           TSR=DTEMP-CAPH
```

```
063           TSS=DTEMP+CAPH
064     C
065           IF(HR.EQ.30.) THEN
066     C       **COMPUTE SOLAR FLUX AT SOLAR NOON
067     C       **TIME OF SOLAR NOON
068             TSN=(TSR+TSS)/2.
069             GMT=TSN+FLOAT(IBTZ)
070             SOLHA=15.*(GMT-EM)-SLONG
071             ALSN=SINLAT*SINCD+COSLAT*COSCD*COS(0.0174533*SOLHA)
072     C       **ECCENTRICITY OF EARTHS ORBIT
073             ECCENT=0.0167
074             PI=3.14159
075             OMEGA=2.*PI/365.
076             RDVCSQ=1./(1.-ECCENT*COS(OMEGA*JULDY))**2
077     C       **SOLAR CONSTANT (WATTS/M**2)
078             SC=1367.
079             A1=SC*RDVCSQ*ALSN
080             ALPHA=ALSN
081             HR=TSN
082           ELSE
083     C       **COMPUTE SIN SOLAR ELEVATION ANGLE AT REQUESTED TIME
084             IF(HR.LE.TSR.OR.HR.GE.TSS) THEN
085               ALPHA=0.0
086             ELSE
087               GMT=HR+FLOAT(IBTZ)
088               SOLHA=15.*(GMT-EM)-SLONG
089               ALPHA=SINLAT*SINCD+COSLAT*COSCD*COS(0.0174533*SOLHA)
090             ENDIF
091           ENDIF
092     C
093           RETURN
094           END
```

```
001     C  SUBROUTINE: MET.SRFC
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO READ IN SURFACE DATA FROM UNIT 2 AND
005     C  CREATE DIRECT ACCESS DISK FILES OF SURFACE WEATHER OBSERVATIONS
006     C  ON UNIT 10.  THE SURFACE DATA MUST BE INPUT GROUPED BY STATION.
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C  THE INPUTS ARE:
009     C    THRU ARGUMENT:
010     C      NSSTA = NUMBER OF SURFACE WEATHER STATIONS.
011     C      DS = SEED NUMBER FOR RANDOM WIND DIRECTION ADJUSTMENT
012     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
013     C                              8(PACIFIC).
014     C      TZNAM = BASE TIME ZONE NAME
015     C      LUO = LOGICAL UNIT TO WHICH THE PRINTOUT IS DIRECTED.
016     C    THRU LOGICAL UNIT 2:
017     C      THE INPUT DATA IS IN TWO MAJOR GROUPINGS.  FIRST IS A SECTION
018     C      CONTAINING A RECORD FOR EACH STATION IN WHICH THE FOLLOWING
019     C      PARAMETERS ARE DEFINED:
020     C      IDS = SURFACE STATION IDENTIFICATION NUMBER.
021     C      NAMES = STATION NAME (A10).
022     C      SCOX = X-COORDINATE OF STATION.
023     C      SCOY = Y-COORDINATE OF STATION.
024     C      ELS = STATION ELEVATION (M. MSL)
025     C      ICODE = CODE SPECIFYING COORDINATE SYSTEM
026     C                1 = LATITUDE-LONGITUDE
027     C                2 = UTM ZONE 12
028     C                3 = UTM ZONE 13
029     C                4 = GRAMA
030     C      NOBS = NUMBER OF OBSERVATIONS FOR STATION.
031     C      ITZS = BASE TIME ZONE OF STATION = 5(EASTERN), 6(CENTRAL),
032     C                                         7(MOUNTAIN), 8(PACIFIC).
033     C
034     C      A SAMPLE RECORD FOR GRAND JUNCTION IS:
035     C      12345678901234567890123456789012345678901234567890123456789012345678901234567890
036     C           1 GRND JUNCT    39.117   108.533  1474.0     1  24   7
037     C
038     C      FOLLOWING THE PARAMETER SPECIFICATION RECORDS ARE THE HOURLY
039     C      OBSERVATIONS.  THE OBSERVATIONS ARE GROUPED BY STATION.  EACH
040     C      RECORD CONTAINS THE FOLLOWING INFORMATION:
041     C      ID = STATION IDENTIFICATION NUMBER.
042     C      ITIME = TIME OF OBSERVATION (MO DA YR HR)
043     C      WD = WIND DIRECTION (DEG)
044     C      WS = WIND SPEED (KNOTS)
045     C      TMP = TEMPERATURE (DEG. F)
046     C      DP = DEW POINT (DEG. F)
047     C      PRES = PRESSURE (MILLIBARS OR INCHES MERCURY)
048     C             IF PRESSURE IN MILLIBARS IMPLIES ACTUAL STATION
049     C             PRESSURE IS INPUT.  IF PRESSURE IN INCHES MERCURY
050     C             IMPLIES ALTIMETER PRESSURE IS GIVEN AND THEREFORE
051     C             SUBROUTINE ALTIM IS CALLED TO CONVERT TO STATION
052     C             PRESSURE.  IF PRESSURE IS EQUAL TO ZERO IMPLIES
053     C             MISSING PRESSURE VALUE.
054     C      ICC = CLOUD COVER IN TENTHS (0 TO 10)
055     C
056     C      A SAMPLE RECORD FOR GRAND JUNCTION IS:
057     C      12345678901234567890123456789012345678901234567890123456789012345678901
058     C           1  08108005  90.0  7.0  68.0  36.0   29.92  3
059     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
060     C  THE OUTPUTS ARE:
061     C    THRU COMMON BLOCK 'SRFC1':
062     C      SCOX = ARRAY OF X-COORD FOR STATIONS (KM)
```

```
063     C       SCOY = ARRAY OF Y-COORD FOR STATIONS (KM)
064     C       ELS = ARRAY OF STATION ELEVATIONS (M MSL)
065     C       IDS = ARRAY OF STATION ID NUMBERS
066     C       NAMES = ARRAY OF STATION NAMES
067     C       NOBS = ARRAY OF NUMBER OF OBSERVATIONS FOR STATIONS
068     C       SLAT = ARRAY OF LATITUDE OF STATIONS (DEG)
069     C       SLONG = ARRAY OF LONGITUDE OF STATIONS (DEG)
070     C       ISSRT = ARRAY OF STARTING RECORD NUMBER OF OBSERVATIONS FOR
071     C                 STATIONS ON DIRECT ACCESS DISK FILE ON UNIT 10
072     C       ISEND = ARRAY OF ENDING RECORD NUMBER OF OBSERVATIONS
073     C     THRU LOGICAL UNIT 10:
074     C       ONE RECORD FOR EACH OBSERVATION PERIOD IS WRITTEN OUT TO THE
075     C       UNFORMATTED DIRECT ACCESS DISK FILE ON UNIT 10.  EACH RECORD
076     C       CONTAINS:
077     C       IDS = SURFACE STATION ID NUMBER
078     C       IJUL = JULIAN DATE OF OBSERVAIATION.
079     C       IHR = HOUR OF OBSERVATION.
080     C       WX = X-COMPONENT OF WINDS (M/S)
081     C       WY = Y-COMPONENT OF WINDS (M/S)
082     C       TMP = TEMPERATURE (DEG C)
083     C       PT = POTENTIAL TEMPERATURE (DEG. K)
084     C       RHO = AIR DENSITY (KG/M**3)
085     C       ICC = CLOUD COVER (TENTHS)
086     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
087     C  FILES READ FROM:
088     C       LOGICAL UNIT 2 = SEQUENTIAL FORMATTED FILE OF SURFACE OBS
089     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
090     C  FILES WRITTEN TO:
091     C       LOGICAL UNIT 10 = DIRECT ACCESS UNFORMATTED DISK FILE OF SURFACE
092     C                         WEATHER OBSERVATIONS
093     C       LOGICAL UNIT LUO = RUN INFO IS OUTPUT
094     C       LOGICAL UNIT  6 = ERROR MESSAGES
095     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
096     C  CALLING SUBROUTINES:
097     C       AAMAIN
098     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
099     C  CALLED SUBROUTINES:
100     C       ALTIM, GRID, JULIAN, LAT, RDM, UTM
101     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
102     C  COMMON BLOCKS:
103     C       SRFC1
104     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
105     C  PARAMETERS DEFINITIONS:
106     C       NNS = NUMBER OF SURFACE WEATHER STATIONS
107     C       LREC = 1*9, [LENGTH OF RECORD IN OUTPUT DISK FILE (WORDS)]
108     C       NREC = TOTAL # OF RECORDS IN OUTPUT DISK FILE.
109     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
110     C
111           SUBROUTINE SRFC (NSSTA,DS.IBTZ,TZNAM,LUO)
112     C
113           PARAMETER
114          +    NNS=15,         NREC=10800,        LREC=9
115     C
116           DIMENSION
117          +    ITZS(NNS)
118     C
119           CHARACTER*10
120          +    NAMES,          TZNAM
121     C
122           DOUBLE PRECISION
123          +    DS
124     C
```

```
      COMMON /SRFC1/
     +    ELS(NNS),      IDS(NNS),      NAMES(NNS),    NOBS(NNS),
     +    SCOX(NNS),     SCOY(NNS),     SLAT(NNS),     SLONG(NNS),
     +    ISSRT(NNS),    ISEND(NNS)
C
C             ***SPECIFY CONSTANTS USED IN DENSITY CALCULATION
      THETA0=300.
      HA=102.4*THETA0
      ROA=353.1/THETA0
C
C             ***FIRST READ IN STATION INFORMATION.
C***************WRITE OUT RUN INFO TO UNIT LUO************************
      WRITE(LUO,40) TZNAM
      WRITE(LUO,41)
C*********************************************************************
      DO 4 J=1,NSSTA
        READ (2,10) IDS(J),NAMES(J),XU,YU,ELS(J),ICODE,NOBS(J),ITZS(J)
        GO TO (2,1,1,3), ICODE
    1   IZONE=ICODE
        CALL UTM (IZONE,XU,YU,SCOX(J),SCOY(J))
        CALL GRID (SCOX(J),SCOY(J),SLAT(J),SLONG(J))
        GO TO 4
    2   SLAT(J)=XU
        SLONG(J)=YU
        CALL LAT (SLAT(J),SLONG(J),SCOX(J),SCOY(J))
        GO TO 4
    3   SCOX(J)=XU
        SCOY(J)=YU
        CALL GRID (SCOX(J),SCOY(J),SLAT(J),SLONG(J))
    4 CONTINUE
C
C             ***DEFINE DIRECT ACCESS DISK FILES.
      DEFINE FILE 10 (NREC,LREC,U,IIW)
      IIW=1
C
C             ***READ IN SURFACE METEOROLOGICAL DATA.
      DO 6 J=1,NSSTA
      DO 5 K=1,NOBS(J)
      READ (2,11) ID,ITIME,WD,WS,TMP,DP,PRES,ICC
      IF (ID.NE.IDS(J))  THEN
        WRITE (6,12)
        STOP 'SRFC'
      ENDIF
      CALL JULIAN (ITIME,ITZS(J),IBTZ,IHR,IJUL)
      IF(K.EQ.1) THEN
        IHRB=IHR
        IJULB=IJUL
      ENDIF
      IHRE=IHR
      IJULE=IJUL
C
C             ***RANDOMIZE WIND DIRECTION IF INPUT ON WHOLE 10 DEGREES.
C             ***THE CORRECTION IS FROM -4 TO 5 DEGREES.
      CKW=REAL(INT(WD/10.))*10.
      IF (WD.EQ.CKW)  THEN
        CALL RDM (DS,OUT)
        WD=WD+OUT
        IF (WD.GT.360.) WD=WD-360.
        IF (WD.LT.0.) WD=360.+WD
      ENDIF
C
C             ***CONVERT TEMPERATURE IN DEGREES F TO DEGREES C.
```

```
      TMP=(TMP-32.)/1.8
      DP=(DP-32.)/1.8
C
C             ***IF PRESSURE IS ALTIMETER SETTING THEN CONVERT TO
C             ***STATION PRESSURE.
      IF (PRES.LT.50..AND.PRES.GT.10.)  THEN
        CALL ALTIM (ELS(J),PRES,PP)
        PRES=PP
      ENDIF
C
C             ***IF PRESSURE NOT GIVEN (PRES=0.) THEN COMPUTE DENSITY
C             ***AND PRESSURE FROM THE DRY ADIABATIC DENSITY
C             ***RELATIONSHIP GIVEN BY DRAKE (AUG 1981) ASSUMING
C             ***THETA = 300 DEG K.
      IF(PRES.EQ.0.) THEN
        RHO=ROA*(1.-ELS(J)/HA)**2.5
        PRES=RHO*(TMP+273.15)/0.3484321
      ELSE
C             ***COMPUTE AIR DENSITY (KG/M**3)
C             ***0.3484321 = (100KG/(M*S**2) PER MB)/(287M**2/(K*S**2))
        RHO=0.3484321*PRES/(TMP+273.15)
      ENDIF
C
C             ***COMPUTE POTENTIAL TEMPERATURE.
      PT=(TMP+273.15)*(1000./PRES)**.286
C
C             ***RESOLVE WINDS INTO COMPONENTS AND CONVERT FROM KNOTS
C             ***TO M/S.
      THET=(270.-WD)*3.14159/180.
      WX=WS*COS(THET)*.5148
      WY=WS*SIN(THET)*.5148
C
C             ***WRITE OUT DATA TO DIRECT ACCESS DISK FILES.
      WRITE (10'IIW) IDS(J),IJUL,IHR,WX,WY,TMP,PT,RHO,ICC
C
    5 CONTINUE
C*****************WRITE OUT RUN INFO TO UNIT LUO************************
      WRITE(LUO,42) NAMES(J),IDS(J),SCOX(J),SCOY(J),SLAT(J),SLONG(J),
     +              ELS(J),NOBS(J),IJULB,IHRB,IJULE,IHRE
C*********************************************************************
    6 CONTINUE
C
C             ***COMPUTE STARTING AND ENDING RECORD FOR STATION.
      ISSRT(1)=1
      ISEND(1)=ISSRT(1)+NOBS(1)-1
      DO 30 J=2,NSSTA
        ISSRT(J)=ISSRT(J-1)+NOBS(J-1)
        ISEND(J)=ISSRT(J)+NOBS(J)-1
   30 CONTINUE
C
      RETURN
   10 FORMAT (I6,1X,A10,2F10.3,F8.1,3I4)
   11 FORMAT (I6,I10,F6.1,F5.1,2F6.1,F8.2,I3)
   12 FORMAT (1H ,' STATION IDS DO NOT MATCH IN SRFC .')
   14 FORMAT (1H ,5X,I3,2X,A10,2X,2F10.3)
   40 FORMAT(1H1,///,5X,'TRACE OF SUBROUTINE SRFC EXECUTION - SRFC',
     +' READS IN ALL SURFACE DATA FROM THE INPUT FILE AND WRITES',
     +/,5X,'OUT DIRECT ACCESS DISK FILES FOR USE BY OTHER PROGRAM',
     +' COMPONENTS.  THE BASE TIME ZONE IS',A10,'.' )
   41 FORMAT(1H0,/,5X,'SUMMARY OF STATIONS PROCESSED:',/,10X,'NAME',
     +6X,'ID NO',2X,'X (KM)',2X,'Y (KM)',2X,'LAT (DEG)',2X,
     +'LONG (DEG)',2X,'ELEV (M MSL)',2X,'NO OBS',2X,'BEG DA',2X,
```

```
249           +'BEG HR',2X,'END DA',2X,'END HR')
250         42 FORMAT(1H ,6X,A10,2X,I6,2X,F6.1,2X,F6.1,3X,F7.3,5X,F7.3,6X,F6.1,
251           +7X,I3,4(5X,I3))
252            END
```

```
001     C  SUBROUTINE: MET.STAB
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO DETERMINE THE P-G-T STABILITY CLASS
005     C  FOR A GIVEN HOUR OF THE DAY USING THE METHOD GIVEN IN MESOPAC.
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C      ISR = TIME OF SUNRISE (1-24 LST)
009     C      ISS = TIME OF SUNSET (1-24 LST)
010     C      IHR = TIME OF DAY FOR WHICH STABILITY IS DESIRED (1-24 LST)
011     C      HCM = HEIGHT OF CONVECTIVE MIXING DEPTH (M AGL)
012     C      WS = WIND SPEED AT 10 M AGL (M/S)
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  THE OUTPUTS ARE:
015     C      ISTAB = P-G-T STABILITY CATEGORY
016     C                  1 - VERY UNSTABLE
017     C                  2 - UNSTABLE
018     C                  3 - SLIGHTLY UNSTABLE
019     C                  4 - NEEUTRAL
020     C                  5 - SLIGHTLY STABLE
021     C                  6 - STABLE
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLING SUBROUTINES:
024     C      MIXSTB
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  CALLED SUBROUTINES:
027     C      (NONE)
028     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
029     C  COMMON BLOCKS:
030     C      (NONE)
031     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
032     C
033           SUBROUTINE STAB (ISR,ISS,IHR,HCM,WS,ISTAB)
034     C
035           DIMENSION
036          +    IDAY(3,5),     NIGHT(5)
037     C
038           DATA NIGHT/6,5,4,4,4/
039           DATA IDAY/1,1,2,1,2,3,2,2,3,3,3,4,3,4,4/
040     C
041     C             ***DETERMINE WIND SPEED CATEGORY.
042           IF (WS.LT.2.) IC=1
043           IF (WS.GE.2..AND.WS.LT.3.) IC=2
044           IF (WS.GE.3..AND.WS.LT.5.) IC=3
045           IF (WS.GE.5..AND.WS.LT.6.) IC=4
046           IF (WS.GE.6.) IC=5
047     C
048     C             ***DETERMINE STABILITY DEPENDING ON TIME OF DAY AND
049     C             ***CONVECTIVE MIXING DEPTH.
050           IF (IHR.LT.ISR.OR.IHR.GT.ISS)  THEN
051             ISTAB=NIGHT(IC)
052           ELSE
053             IF (HCM.LT.500.) ISTAB=IDAY(3,IC)
054             IF (HCM.GE.500..AND.HCM.LT.1000.) ISTAB=IDAY(2,IC)
055             IF (HCM.GE.1000.) ISTAB=IDAY(1,IC)
056           ENDIF
057     C
058           RETURN
059           END
```

```
001     C  SUBROUTINE: MET.SUBA0
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE DAILY AVERAGE RATIO (A0) OF
005     C  SENSIBLE HEAT FLUX TO EXTRATERRESTRIAL SOLAR IRRADIANCE ON A
006     C  HORIZONTAL SURFACE AT EACH STATION HAVING SURFACE AND UPPER AIR
007     C  WEATHER OBSERVATIONS.  THE A0 IS THEN INTERPOLATED TO THE SURFACE
008     C  AND UPPER AIR STATIONS NOT HAVING UPPER AIR OR SURFACE OBSERVATIONS,
009     C  RESPECTIVELY.  THE OUTPUT IS A DAILY AVERAGE A0 AT EACH SURFACE
010     C  STATION LOCATION, AND AT EACH UPPER AIR STATION LOCATION.  THE
011     C  APPROACH IS TO INTEGRATE THE HEAT ENERGY BETWEEN THE MORNING AND
012     C  AFTERNOON POTENTIAL TEMPERATURE SOUNDINGS AND DIVIDE THAT BY THE
013     C  TOTAL INCOMING SOLAR ENERGY FROM SUNRISE TO SUNSET.  THE TOTAL
014     C  INCOMING SOLAR ENERGY IS DETERMINED BY INTEGRATING UNDER THE SOLAR
015     C  FLUX CURVE TAKING INTO ACCOUNT HOURLY VARIATIONS IN CLOUD COVER.
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  THE INPUTS ARE:
018     C    THRU ARGUMENT:
019     C      JUL = JULIAN DAY
020     C      NSTA = NUMBER OF UPPER AIR STATIONS
021     C      NSSTA = NUMBER OF SURFACE STATIONS
022     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
023     C                              8(PACIFIC)
024     C      RAD = MAXIMUM SEPARATION DISTANCE, BETWEEN A SURFACE OR 'OTHER'
025     C            STATION AND AN UPPER AIR STATION, AT WHICH FOR ANY ACTUAL
026     C            SEPARATION DISTANCE GREATER THAN 'RAD', A 1/R**2
027     C            INTERPOLATION OF THE SOUNDING TO THE SURFACE OR 'OTHER'
028     C            STATION WILL NOT BE PERFORMED (KM).
029     C      A01ST = LOGICAL VARIABLE IF .TRUE. MEANS FIRST CALL TO SUBA0
030     C      IA0 = 1  [COMPUTE A0 AT UPPER AIR STATION LOCATIONS]
031     C            2  [COMPUE A0 AT SURFACE STATION LOCATIONS]
032     C    THRU COMMON BLOCK 'UPPER1':
033     C      COX = ARRAY OF X-COORD OF UPPER AIR STATIONS (KM)
034     C      COY = ARRAY OF Y-COORD OF UPPER AIR STATIONS (KM)
035     C      IDU = ARRAY OF UPPER AIR STATIONS IDENTIFICATIONS
036     C      XLAT = ARRAY OF STATION LATITUDES (DEG)
037     C      YLONG = ARRAY OF STATION LONGITUDES (DEG)
038     C      IUSRTT = ARRAY OF STARTING RECORD NUMBER OF TEMP SNDGS FOR ALL
039     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 8
040     C      IUENDT = ARRAY OF ENDING RECORD NUMBER OF TEMP SNDGS FOR ALL
041     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 8
042     C      ITZU = ARRAY OF TIME ZONES WHERE STATIONS LOCATED
043     C    THRU COMMON BLOCK 'SRFC1':
044     C      NOBS = NUMBER OF SURFACE OBSERVATIONS FOR EACH STATION
045     C      SCOX = ARRAY OF X-COORD OF SURFACE STATIONS (KM)
046     C      SCOY = ARRAY OF Y-COORD OF SURFACE STATIONS (KM)
047     C      ISSRT = ARRAY OF STARTING RECORD NUMBER OF SRFC OBS FOR ALL
048     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
049     C    THRU COMMON BLOCK 'UPSFC1':
050     C      NMSTA = NUMBER OF STATIONS WITH BOTH SURFACE AND UPPER AIR OBS
051     C      IUM = ARRAY OF INDEX NUMBERS OF UPPER AIR STATIONS WITH SURFACE
052     C            OBSERVATIONS ALSO
053     C      ISM = ARRAY OF INDEX NUMBERS OF SURFACE STATIONS WITH UPPER AIR
054     C            OBSERVATIONS ALSO
055     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
056     C  THE OUTPUTS ARE:
057     C    THRU THE ARGUMENT:
058     C      A0U = DAILY AVERAGE A0 FOR ALL UPPER AIR STATIONS
059     C      A0S = DAILY AVERAGE A0 FOR ALL SURFACE STATIONS
060     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
061     C  CONSTANTS DEFINITIONS:
062     C      A0SUM = .59  [MAX FRACTION OF EXTRA-TERRESTRIAL SOLAR RAD AS
```

```
063     C                       SENSIBLE HEAT DURING THE SUMMER]
064     C       A0WIN = .53  [MAX FRACTION OF EXTRA-TERRESTRIAL SOLAR RAD AS
065     C                       SENSIBLE HEAT DURING THE WINTER]
066     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
067     C  CALLING SUBROUTINES:
068     C       BLPARS, COEFF
069     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
070     C  CALLED SUBROUTINES:
071     C       FINDU, READS, SOLAR, WARM
072     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
073     C  COMMON BLOCKS:
074     C       A01, UPPER1, SRFC1, UPSFC1
075     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
076     C  PARAMETERS DEFINITIONS:
077     C       NNS = NUMBER OF SURFACE STATIONS
078     C       NOB = NUMBER OF OBSERVATIONS IN A SOUNDING
079     C       NUA = NUMBER OF UPPER AIR STATIONS
080     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
081     C
082           SUBROUTINE SUBA0(JUL,NSTA,NSSTA,IBTZ,RAD,A01ST,IA0,A0U,A0S)
083     C
084           PARAMETER
085          +     NNS=15,          NOB=30,          NUA=10
086     C
087           LOGICAL
088          +     MORN,            AFT,             RDS1ST,          PASS,
089          +     A01ST
090     C
091           CHARACTER*10
092          +     NAME,            NAMES
093     C
094           DIMENSION
095          +     BLA(NOB),        A0U(NUA),        A0S(NNS),        BETA(11),
096          +     PTA(NOB),        PTM(NOB),        THTA(NOB),       THTMP(NOB),
097          +     THTM(NOB),       THTAP(NOB)
098     C
099           COMMON /UPPER1/
100          +     COX(NUA),        COY(NUA),        EL(NUA),         IDU(NUA),
101          +     NAME(NUA),       NSDT(NUA),       NSDW(NUA),       XLAT(NUA),
102          +     YLONG(NUA),      IUSRTW(NUA),     IUSRTT(NUA),     IUENDW(NUA),
103          +     IUENDT(NUA),     ITZU(NUA)
104     C
105           COMMON /SRFC1/
106          +     ELS(NNS),        IDS(NNS),        NAMES(NNS),      NOBS(NNS),
107          +     SCOX(NNS),       SCOY(NNS),       SLAT(NNS),       SLONG(NNS),
108          +     ISSRT(NNS),      ISEND(NNS)
109     C
110           COMMON /UPSFC1/
111          +     NMSTA,           IUM(NUA),        ISM(NNS)
112     C
113           COMMON /A01/
114          +     IUPNT(NUA),      NOBSS(NNS)
115     C
116     C            BETA IS THE SOLAR RADIATION REDUCTION FACTOR
117           DATA BETA/1.0,.91,.84,.79,.75,.72,.68,.62,.53,.41,.23/
118     C
119     C               ***CONSTANTS
120           A0SUM=0.59
121           A0WIN=0.53
122           ICLKK=(JUL-1)*24
123     C
124           IF(A01ST) THEN
```

```
125           DO 1 L=1,NSTA
126             IUPNT(L)=IUSRTT(L)-1
127         1   CONTINUE
128           ENDIF
129     C
130           DO 2 L=1,NSTA
131             A0U(L)=999.
132         2 CONTINUE
133     C
134           DO 3 L=1,NSSTA
135             A0S(L)=999.
136         3 CONTINUE
137     C
138     C               ***LOOP ON STATIONS
139           RDS1ST=.FALSE.
140           NS=0
141           DO 10 L=1,NMSTA
142           IF(A01ST) RDS1ST=.TRUE.
143     C
144     C               ***COMPUTE TIME OF SUNRISE AND SUNSET
145           TSN=30.
146           CALL SOLAR(JUL,TSN,XLAT(IUM(L)),YLONG(IUM(L)),ITZU(IUM(L)),A1,
147          +           ALPHA,TSR,TSS)
148           ISR=NINT(TSR)+(ITZU(IUM(L))-IBTZ)
149           ISS=ISR+NINT(TSS-TSR)
150     C
151     C               *FIND MORNING SOUNDING.
152           MORN=.TRUE.
153           AFT=.FALSE.
154           CALL FINDU(JUL,ISR,ISS,IDU(IUM(L)),IUPNT(IUM(L)),IUENDT(IUM(L)),
155          +           MORN,AFT,IHRM,HMM,NOBM,THTMP,THTM,BLA,PTM)
156           IF(.NOT.MORN) GO TO 10
157     C
158     C               *FIND AFTERNOON SOUNDING
159           MORN=.FALSE.
160           AFT=.TRUE.
161           CALL FINDU(JUL,ISR,ISS,IDU(IUM(L)),IUPNT(IUM(L)),IUENDT(IUM(L)),
162          +           MORN,AFT,IHRA,HMA,NOBA,THTAP,THTA,BLA,PTA)
163           IF(.NOT.AFT) GO TO 10
164     C
165     C               **COMPUTE A0 FROM DIFFERENCE BETWEEN MORNING AND
166     C               **AFTERNOON SOUNDINGS.  FIRST COMPUTE EXTRA TERRESTRIAL
167     C               **SOLAR RADIATION AT STATION BY INTEGRATING UNDER THE
168     C               **SOLAR RADIATION CURVE USING .5 HR TIME STEPS.
169           EXTER=0.
170           TIM=TSR
171           TOT=A1/ALPHA
172           DO 5 JJ=1,40
173             TIM=TIM+.5
174             ICLK=NINT(TIM)+ICLKK
175             CALL READS (ICLK,NOBSS(ISM(L)),ISSRT(ISM(L)),NOBS(ISM(L)),
176          +              ISM(L),RDS1ST,PASS,BL,BL,BL,BL,ICCI)
177             RDS1ST=.FALSE.
178             IF (PASS) GO TO 10
179             CALL SOLAR(JUL,TIM,XLAT(IUM(L)),YLONG(IUM(L)),ITZU(IUM(L)),A1,
180          +             ALPHA,TTSR,TTSS)
181             EXTER=TOT*BETA(ICCI+1)*ALPHA*1800.+EXTER
182             IF(TIM.GT.TSS) GO TO 6
183         5 CONTINUE
184         6 CONTINUE
185     C               *COMPUTE SENSIBLE HEAT INPUT BETWEEN MORNING AND
186     C               *AFTERNOON SOUNDINGS.
```

```
      CALL WARM(NOBM,NOBA,THTM,THTA,THTMP,THTAP,PTM,PTA,ENER)
      A0=ENER/EXTER
C             *A0 NOT GREATER THAN 0.59 IN SUMMER, OR 0.53 IN
C             *WINTER.
      A0MAX=A0WIN+(A0SUM-A0WIN)*SIN(FLOAT(JUL)/366.*3.14159)
      IF(A0.GT.A0MAX) A0=A0MAX
      IF(A0.LT.0.0001) A0=0.0001
      A0U(IUM(L))=A0
      A0S(ISM(L))=A0
      NS=NS+1
C
   10 CONTINUE
C
      IF(NS.EQ.0) THEN
        WRITE(6,50) JUL
        STOP 'SUBA0'
      ENDIF
C
C             ***1/R**2 WEIGHT A0 TO OTHER UPPER AIR STATIONS
      IF(IA0.EQ.1) THEN
      DO 13 J=1,NSTA
        IF(A0U(J).NE.999.) GO TO 13
        WF=0.
        A0US=0.
        DO 12 L=1,NMSTA
          IF(A0U(IUM(L)).EQ.999.) GO TO 12
          RDIF=SQRT((COX(IUM(L))-COX(J))**2+(COY(IUM(L))-COY(J))**2)
          IF (RDIF.LE.RAD)  THEN
            IF (RDIF.LE..1) RDIF=.001
            REC=1./RDIF**2
            WF=REC+WF
            A0US=REC*A0U(IUM(L))+A0US
          ENDIF
   12   CONTINUE
        A0U(J)=A0US/WF
   13 CONTINUE
      ENDIF
C
C             ***1/R**2 WEIGHT A0 TO OTHER SURFACE STATIONS
      IF(IA0.EQ.2) THEN
      DO 16 J=1,NSSTA
        IF(A0S(J).NE.999.) GO TO 16
        WF=0.
        A0SS=0.
        DO 15 L=1,NMSTA
          IF(A0S(ISM(L)).EQ.999.) GO TO 15
          RDIF=SQRT((SCOX(ISM(L))-SCOX(J))**2+(SCOY(ISM(L))-SCOY(J))**2)
          IF (RDIF.LE.RAD)  THEN
            IF (RDIF.LE..1) RDIF=.001
            REC=1./RDIF**2
            WF=REC+WF
            A0SS=REC*A0S(ISM(L))+A0SS
          ENDIF
   15   CONTINUE
        A0S(J)=A0SS/WF
   16 CONTINUE
      ENDIF
C
      RETURN
   50 FORMAT(1H ,5X,'NO STATIONS TO COMPUTE A0 ON JULIAN DAY',I4)
      END
```

```
001     C  SUBROUTINE: MET.TEMP
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO FILL VERTICAL AND SURFACE TEMPERATURE
005     C  ARRAYS AND VERTICAL POT TEMP AND SURFACE ADJUSTED POT TEMP ARRAYS
006     C  USED TO COMPUTE THE POTENTIAL TEMPERATURE LAPSE RATE AND THE BULK
007     C  TEMPERATURE IN APPROACH FLOWS TO TERRAIN OBSTACLES IN THE FROUDE
008     C  NUMBER FLOW MODIFICATION SCHEME.  THE SURFACE ARRAYS ARE FILLED
009     C  FROM SUNRISE ON DAY 'JUL' TO SUNRISE ON DAY 'JUL+1'.  THE MORNING
010     C  AND AFTERNOON SOUNDINGS ON DAY 'JUL' ARE DETERMINED AND THE
011     C  MORNING SOUNDING ON DAY 'JUL+1' IS DETERMINED.  ONLY STATIONS HAVING
012     C  BOTH UPPER AIR AND SURFACE DATA ARE USED.  THE SURFACE POTENTIAL
013     C  TEMPERATURE VALUES FROM SUNRISE TO SUNSET ON DAY 'JUL' ARE ADJUSTED
014     C  ACCORDING TO WARM OR COLD AIR ADVECTION.  THIS ADVECTION RATE
015     C  (DEGK/HR) IS COMPUTED FROM THE DIFFERENCE BETWEEN THE MORNING AND
016     C  AFTERNOON SOUNDING AT 3000 MAGL.
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE INPUTS ARE:
019     C    THRU ARGUMENT:
020     C      JUL = JULIAN DATE OF DAY BEING PROCESSES
021     C      IHRR = HOUR OF DAY BEING PROCESSED (1 TO 24)
022     C      NSTA = NUMBER OF UPPER AIR STATIONS
023     C      NSSTA = NUMBER OF SURFACE WEATHER STATIONS
024     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
025     C                              8(PACIFIC)
026     C      TMP1ST = LOGICAL VARIABLE IF .TRUE. MEANS FIRST CALL TO
027     C               SUBROUTINE 'TEMP' AND CERTAIN INITALIZATION STEPS
028     C               ARE TAKEN
029     C    THRU LOGICAL UNIT 10:
030     C      IDD = ID OF SURFACE OBS. STATION USED TO FILL HOURLY TEMP
031     C            ARRAYS FROM SUNRISE ON DAY 'JUL' TO SUNRISE ON DAY 'JUL+1'
032     C      IJUL = JULIAN DATE OF SURFACE OBSERVATION
033     C      IHR = HOUR OF DAY OF SURFACE OBSERVATION (1 TO 24)
034     C      TMPP = SURFACE OBSERVATION TEMPERATURE (DEG C)
035     C      PTT = SURFACE OBSERVATION POTENTIAL TEMPERATURE (DEG K)
036     C    THRU COMMON BLOCK 'UPPER1':
037     C      IDU = ARRAY OF UPPER AIR STATIONS IDENTIFICATIONS
038     C      IUSRTT = ARRAY OF STARTING RECORD NUMBER OF TEMP SNDGS FOR ALL
039     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 9
040     C      IUENDT = ARRAY OF ENDING RECORD NUMBER OF TEMP SNDGS FOR ALL
041     C               UPPER AIR STNS ON DIRECT ACCESS DISK FILE ON UNIT 9
042     C      XLAT = ARRAY OF UPPER AIR STATIONS LATITUDES (DEG)
043     C      YLONG = ARRAY OF UPPER AIR STATIONS LONGITUDES (DEG)
044     C    THRU COMMON BLOCK 'SRFC':
045     C      ISSRT = ARRAY OF STARTING RECORD NUMBER OF SRFC OBS FOR ALL
046     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
047     C      ISEND = ARRAY OF ENDING RECORD NUMBER OF SRFC OBS FOR ALL
048     C              SRFC STNS ON DIRECT ACCESS DISK FILE ON UNIT 10
049     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
050     C  THE OUTPUTS ARE:
051     C    THRU ARGUMENT:
052     C      TMP1ST = LOGICAL VARIABLE SET .FALSE. TO INDICATE THE FIRST
053     C               CALL TO THIS SUBROUTINE HAS BEEN MADE
054     C    THRU COMMON BLOCK 'TEMP1':
055     C      IS = ARRAY OF INDEX NUMBERS OF SURFACE STATIONS USED
056     C      IU = ARRAY OF INDEX NUMBERS OF UPPER AIR STATIONS USED
057     C      JULI = ARRAY OF JULIAN DATE FOR EACH STATION INDICATING THE DAY
058     C             ON WHICH THE CURRENT UPPER AIR AND SURFACE DATA IS TO BE
059     C             USED.  WHEN JUL = JULI+1 THEN NEW DATA IS READ IN FOR THE
060     C             STATION.
061     C      IUPNT = ARRAY GIVING THE CURRENT RECORD NUMBER FOR EACH STATION
062     C              TO BE READ FROM THE UPPER AIR DIRECT ACCESS DISK FILE.
```

```
063     C       ISPNT = ARRAY GIVING THE CURRENT RECORD NUMBER TO BE READ FROM
064     C               THE SURFACE WEATHER DATA DIRECT ACCESS DISK FILE.
065     C       IUENDR = ARRAY GIVING THE ENDING RECORD NUMBER FOR EACH STATION
066     C                OF UPPER AIR DATA ON THE DIRECT ACCESS DISK FILE.
067     C       ISENDR = ARRAY OF THE ENDING RECORD NUMBER FOR EACH STATION OF
068     C                SURFACE DATA ON THE DIRECT ACCESS DISK FILE.
069     C       CNGTIM = ARRAY OF A LOGICAL VARIABLE FOR EACH STATION IF .TRUE.
070     C                INDICATING THAT THE CURRENT SIMULATION HOUR (IHRR) IS
071     C                ON DAY 'JUL+1' BUT BEFORE SUNRISE.  THEREFORE 24 MUST
072     C                SUBTRACTED FROM THE HOUR-OF-DAY ARRAY INDEX IN THE
073     C                SURFACE DATA ARRAYS TO GET THE CORRECT HOUR.
074     C       DAYFL = ARRAY OF A LOGICAL VARIABLE FOR EACH STATION IF .TRUE.
075     C               INDICATING DATA AVAILABLE FROM SUNRISE TO SUNSET ON
076     C               DAY 'JUL'.
077     C       NTFL = ARRAY OF A LOGICAL VARIABLE FOR EACH STATION IF .TRUE.
078     C              INDICATING DATA AVAILABLE FROM SUNSET ON DAY 'JUL' TO
079     C              SUNRISE ON DAY 'JUL+1'.
080     C     THRU COMMON BLOCK 'LAPSE1':
081     C       ISRS = ARRAY OF SUNRISE TIME FOR EACH STATION ON DAY 'JUL'.
082     C       ISSS = ARRAY OF SUNSET TIME FOR EACH STATION ON DAY 'JUL'.
083     C       NTS = NUMBER OF STATIONS WITH BOTH UPPER AIR AND SURFACE DATA.
084     C       NOBM = ARRAY OF NUM. OF OBS. FOR MORN. SND. FOR STATIONS
085     C       THTM = ARRAY OF OBS. HEIGHTS FOR MORN. SND. FOR STATIONS (M MSL)
086     C       TMPM = ARRAY OF TEMP. OBS. FOR MORN. SND. FOR STATIONS (DEG C)
087     C       PTM = ARRAY OF POT. TEMP. OBS. FOR MORN SND FOR STATNS (DEG K)
088     C       NOBA = --------I
089     C       THTA =         I----  SAME AS ABOVE EXCEPT FOR AFTERNOON
090     C       TMPA =         I      SOUNDING ON DAY 'JUL'
091     C       PTA = ---------I
092     C       NOBM1 = ---------I
093     C       THTM1 =          I----  SAME AS ABOVE EXCEPT FOR MORNING
094     C       TMPM1 =          I      SOUNDING ON DAY 'JUL+1'
095     C       PTM1 = ----------I
096     C       NOBMM = ----I       SAME AS MORNING SOUNDING ON DAY 'JUL' GIVEN
097     C       THTMM =     I----   ABOVE EXCEPT LOWER PART OF POT TEMP SND HAS
098     C       PTMM = -----I       BEEN MODIFIED TO MATCH ADJUSTED SRFC POT TMP
099     C       APT = ARRAY OF HOURLY ADJUSTED SRFC POT TEMP VALUES STARTING AT
100     C             SUNRISE ON DAY 'JUL' AND GOING TO SUNRISE ON DAY 'JUL+1'
101     C       ATMP = ARRAY OF HOURLY SRFC TEMP VALUES STARTING AT SUNRISE ON
102     C              DAY 'JUL' AND GOING TO SUNRISE ON DAY 'JUL+1'
103     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
104     C  CONSTANTS DEFINITIONS:
105     C       ADHT = 3000.  [HEIGHT ABOVE GROUND IN A TEMPERATURE SOUNDING
106     C                      AT WHICH THE WARM OR COLD AIR ADVECTION RATE
107     C                      WILL BE COMPUTED]
108     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
109     C  FILES READ FROM:
110     C       LOGICAL UNIT 10 = DIRECT ACCESS DISK FILE, OF SURFACE WEATHER
111     C                         OBSERVATIONS, CREATED BY SUBROUTINE 'SRFC'.
112     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
113     C  FILES WRITTEN TO:
114     C       LOGICAL UNIT 6 = ERROR MESSAGES
115     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
116     C  CALLING SUBROUTINES:
117     C       FROUDE
118     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
119     C  CALLED SUBROUTINES:
120     C       FINDU, SOLAR, TPROF
121     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
122     C  COMMON BLOCKS:
123     C       UPPER1, SRFC1, TEMP1, LAPSE1
124     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
C  PARAMETERS DEFINITIONS:
C       NNS = NUMBER OF SURFACE STATIONS
C       NUA = NUMBER OF UPPER AIR STATIONS
C       NOB = NUMBER OF OBSERVATIONS IN AN UPPER AIR SOUNDING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE TEMP(JUL,IHRR,NSTA,NSSTA,IBTZ,TMP1ST)
C
      PARAMETER
     +    NNS=15,          NOB=30,          NUA=10
C
      LOGICAL
     +    CNGTIM,          DAYFL,           NTFL,            TMP1ST,
     +    MORN,            AFT
C
      DIMENSION
     +    PPT(NOB),        TH(NOB),         TTMP(NOB),       BLA(NOB)
C
      CHARACTER*10
     +    NAME,            NAMES
C
      COMMON /UPPER1/
     +    COX(NUA),        COY(NUA),        EL(NUA),         IDU(NUA),
     +    NAME(NUA),       NSDT(NUA),       NSDW(NUA),       XLAT(NUA),
     +    YLONG(NUA),      IUSRTW(NUA),     IUSRTT(NUA),     IUENDW(NUA),
     +    IUENDT(NUA),     ITZU(NUA)
C
      COMMON /SRFC1/
     +    ELS(NNS),        IDS(NNS),        NAMES(NNS),      NOBS(NNS),
     +    SCOX(NNS),       SCOY(NNS),       SLAT(NNS),       SLONG(NNS),
     +    ISSRT(NNS),      ISEND(NNS)
C
      COMMON /TEMP1/
     +    IS(NUA),         ISENDR(NUA),     ISPNT(NUA),      IU(NUA),
     +    IUENDR(NUA),     IUPNT(NUA),      JULI(NUA),       CNGTIM(NUA),
     +    DAYFL(NUA),      NTFL(NUA)
C
      COMMON /LAPSE1/
     +    APT(36,NUA),     ATMP(36,NUA),    DTDZCK(NUA),     ISRS(NUA),
     +    ISSS(NUA),       NOBA(NUA),       NOBM(NUA),       NOBMM(NUA),
     +    NOBM1(NUA),      NTS,             PTA(NOB,NUA),    PTM(NOB,NUA),
     +    PTMM(NOB,NUA),   PTM1(NOB,NUA),   THTA(NOB,NUA),   THTAP(NOB,NUA),
     +    THTM(NOB,NUA),   THTMM(NOB,NUA),  THTM1(NOB,NUA),  TMPA(NOB,NUA),
     +    TMPM(NOB,NUA),   TMPM1(NOB,NUA)
C
C
C           ***CONSTANTS
      ADHT=3000.
C
C           ***DETERMINE STATIONS TO BE USED IN COMPUTING DTDZ AND
C           ***BULK TEMPERATURE.  THE STATIONS MUST HAVE UPPER
C           ***AIR AND SURFACE OBSERVATIONS.
      IF (TMP1ST) THEN
        TMP1ST=.FALSE.
        IC=0
        DO 93 J=1,NSTA
        DO 93 K=1,NSSTA
        IF (IDU(J).EQ.IDS(K))  THEN
          IC=IC+1
          IU(IC)=J
          IS(IC)=K
          IUPNT(IC)=IUSRTT(J)-1
```

```
187               IUENDR(IC)=IUENDT(J)
188               ISPNT(IC)=ISSRT(K)-1
189               ISENDR(IC)=ISEND(K)
190             ENDIF
191          93 CONTINUE
192             NTS=IC
193             IF (NTS.EQ.0)  THEN
194               WRITE (6,51)
195               STOP 'TEMP'
196             ENDIF
197             DO 499 L=1,NTS
198             JULI(L)=JUL-1
199             TSN=30.
200             CALL SOLAR(JUL,TSN,XLAT(IU(L)),YLONG(IU(L)),IBTZ,Al,ALPHA,
201           +                TSR,TSS)
202             ISRS(L)=NINT(TSR)+(ITZU(IU(L))-IBTZ)
203             ISSS(L)=ISRS(L)+NINT(TSS-TSR)
204          499 CONTINUE
205           ENDIF
206          C
207          C              ***LOOP ON STATIONS
208           DO 500 L=1,NTS
209          C
210           IF(JUL.EQ.JULI(L)+1) THEN
211             IF(IHRR.LT.ISRS(L)) THEN
212               CNGTIM(L)=.TRUE.
213               GO TO 500
214             ENDIF
215          C              *COMPUTE TIME OF SUNRISE AND SUNSET FOR EACH STATION.
216             TSN=30.
217             CALL SOLAR(JUL,TSN,XLAT(IU(L)),YLONG(IU(L)),IBTZ,Al,ALPHA,
218           +                TSR,TSS)
219             ISR=NINT(TSR)+(ITZU(IU(L))-IBTZ)
220             ISS=ISR+NINT(TSS-TSR)
221             ISRS(L)=ISR
222             ISSS(L)=ISS
223             JULI(L)=JUL
224             CNGTIM(L)=.FALSE.
225           ELSE
226             GO TO 500
227           ENDIF
228          C              *FIND MORNING SOUNDING.
229           MORN=.TRUE.
230           AFT=.FALSE.
231           CALL FINDU(JUL,ISR,ISS,IDU(IU(L)),IUPNT(L),IUENDR(L),MORN,AFT,
232           +                IHRM,HMM,NOBM(L),BLA,TH,TTMP,PPT)
233           IF(MORN) THEN
234             DAYFL(L)=.TRUE.
235             DO 40 K=1,NOBM(L)
236               THTM(K,L)=TH(K)
237               TMPM(K,L)=TTMP(K)
238               PTM(K,L)=PPT(K)
239          40   CONTINUE
240           ELSE
241             DAYFL(L)=.FALSE.
242           ENDIF
243          C              *FIND AFTERNOON SOUNDING
244           MORN=.FALSE.
245           AFT=.TRUE.
246           CALL FINDU(JUL,ISR,ISS,IDU(IU(L)),IUPNT(L),IUENDR(L),MORN,AFT,
247           +                IHRA,HMA,NOBA(L),BLA,TH,TTMP,PPT)
248           IF(AFT) THEN
```

```
249             NTFL(L)=.TRUE.
250             DO 41 K=1,NOBA(L)
251               THTA(K,L)=TH(K)
252               TMPA(K,L)=TTMP(K)
253               PTA(K,L)=PPT(K)
254          41   CONTINUE
255           ELSE
256             DAYFL(L)=.FALSE.
257             NTFL(L)=.FALSE.
258             GO TO 500
259           ENDIF
260     C             *FIND MORNING SOUNDING FOR NEXT DAY.
261           JULM=JUL+1
262           MORN=.TRUE.
263           AFT=.FALSE.
264           CALL FINDU(JULM,ISR,ISS,IDU(IU(L)),IUPNT(L),IUENDR(L),MORN,AFT,
265          +           IHRM1,HMM1,NOBM1(L),BLA,TH,TTMP,PPT)
266           IF(MORN) THEN
267             DO 42 K=1,NOBM1(L)
268               THTM1(K,L)=TH(K)
269               TMPM1(K,L)=TTMP(K)
270               PTM1(K,L)=PPT(K)
271          42   CONTINUE
272           ELSE
273             NTFL(L)=.FALSE.
274             IF(.NOT.DAYFL(L)) GO TO 500
275           ENDIF
276     C
277     C             ***FIND SURFACE OBSERVATIONS FOR THIS STATION.
278           DO 6 M=1,36
279           APT(M,L)=999.
280           ATMP(M,L)=999.
281         6 CONTINUE
282     C
283         7 ISPNT(L)=ISPNT(L)+1
284           IF (ISPNT(L).LE.ISENDR(L)) THEN
285             READ (10'ISPNT(L)) IDD,IJUL,IHR,BL,BL,TMPP,PTT,BL,IBL
286             IF (IDD.NE.IDS(IS(L)))  THEN
287               WRITE (6,33)
288               STOP 'TEMP'
289             ENDIF
290             IF (IJUL.LT.JUL) GO TO 7
291             IF(IJUL.GT.JUL) THEN
292               ISPNT(L)=ISPNT(L)-1
293               IUPNT(L)=IUPNT(L)-1
294               DAYFL(L)=.FALSE.
295               NTFL(L)=.FALSE.
296               GO TO 500
297             ELSE
298               APT(IHR,L)=PTT
299               ATMP(IHR,L)=TMPP
300          8    ISPNT(L)=ISPNT(L)+1
301               IF (ISPNT(L).LE.ISENDR(L)) THEN
302                 READ (10'ISPNT(L)) IDD,IJUL,IHR,BL,BL,TMPP,PTT,BL,IBL
303                 IF (IDD.NE.IDS(IS(L)))  THEN
304                   WRITE (6,33)
305                   STOP 'TEMP'
306                 ENDIF
307                 IF (IJUL.GT.JUL)  THEN
308                   ISPNT(L)=ISPNT(L)-1
309                 ELSE
310                   APT(IHR,L)=PTT
```

```
311                 ATMP(IHR,L)=TMPP
312                 GO TO 8
313               ENDIF
314             ELSE
315               IF(IHR.LT.ISS) THEN
316                 DAYFL(L)=.FALSE.
317                 NTFL(L)=.FALSE.
318                 GO TO 500
319               ELSE
320                 NTFL(L)=.FALSE.
321                 IF(.NOT.DAYFL(L)) GO TO 500
322               ENDIF
323             ENDIF
324           ENDIF
325         ELSE
326           DAYFL(L)=.FALSE.
327           NTFL(L)=.FALSE.
328           GO TO 500
329         ENDIF
330   C
331   C               ***FIND SURFACE OBSERVATIONS FOR THIS STATION FOR NEXT
332   C               ***DAY UP TO ISR+2.
333         IF(NTFL(L)) THEN
334           ICNTS=1
335           ISPNT(L)=ISPNT(L)+1
336           IF (ISPNT(L).LE.ISENDR(L)) THEN
337             READ (10'ISPNT(L)) IDD,IJUL,IHR,BL,BL,TMPP,PTT,BL,IBL
338             IF (IDD.NE.IDS(IS(L)))  THEN
339               WRITE (6,33)
340               STOP 'TEMP'
341             ENDIF
342             IF(IJUL.GT.JUL+1.OR.IHR.GT.ISR+2) THEN
343               ISPNT(L)=ISPNT(L)-1
344               NTFL(L)=.FALSE.
345               IF(.NOT.DAYFL(L)) GO TO 500
346             ELSE
347               APT(IHR+24,L)=PTT
348               ATMP(IHR+24,L)=TMPP
349       88      ISPNT(L)=ISPNT(L)+1
350               IF (ISPNT(L).LE.ISENDR(L)) THEN
351                 READ (10'ISPNT(L)) IDD,IJUL,IHR,BL,BL,TMPP,PTT,BL,IBL
352                 ICNTS=ICNTS+1
353                 IF (IDD.NE.IDS(IS(L)))  THEN
354                   WRITE (6,33)
355                   STOP 'TEMP'
356                 ENDIF
357                 IF (IJUL.GT.JUL+1.OR.IHR.GT.ISR+2)  THEN
358                   ISPNT(L)=ISPNT(L)-ICNTS
359                 ELSE
360                   APT(IHR+24,L)=PTT
361                   ATMP(IHR+24,L)=TMPP
362                   GO TO 88
363                 ENDIF
364               ELSE
365                 NTFL(L)=.FALSE.
366                 IF(.NOT.DAYFL(L)) GO TO 500
367               ENDIF
368             ENDIF
369           ELSE
370             NTFL(L)=.FALSE.
371             IF(.NOT.DAYFL(L)) GO TO 500
372           ENDIF
```

```
373           ENDIF
374     C
375     C               ***CHECK TO SEE IF SURFACE POTENTIAL TEMPERATURE ARRAY
376     C               ***HAS AT LEAST 1 VALUE BEFORE SUNRISE, AT LEAST 1
377     C               ***VALUE AFTER SUNSET, AT LEAST 1 VALUE DURING
378     C               ***MIDDAY (12 TO 15) AND AT LEAST 1 VALUE AT SURRISE THE
379     C               ***NEXT DAY.  IF NOT GO TO NEXT STATION.
380     C
381     C               **TEST AFTERNOON PERIOD
382           DO 10 K=ISS,ISS+2
383           IF (APT(K,L).NE.999.) GO TO 90
384        10 CONTINUE
385           DAYFL(L)=.FALSE.
386           NTFL(L)=.FALSE.
387           GO TO 500
388     C
389     C                **TEST MORNING AND MIDDAY PERIOD
390        90 IF(DAYFL(L)) THEN
391             DO 11 J=ISR-4,ISR
392             IF (APT(J,L).NE.999.)  THEN
393               DO 417 M=12,15
394               IF (APT(M,L).NE.999.) THEN
395                 IF(NTFL(L)) GO TO 91
396                 GO TO 12
397               ENDIF
398       417     CONTINUE
399               DAYFL(L)=.FALSE.
400               IF(NTFL(L)) GO TO 91
401               GO TO 500
402             ENDIF
403        11   CONTINUE
404             DAYFL(L)=.FALSE.
405             IF(NTFL(L)) GO TO 91
406             GO TO 500
407           ENDIF
408     C
409     C               **TEST THE NEXT MORNING
410        91 DO 414 N=ISR+20,ISR+26
411           IF(APT(N,L).NE.999.) GO TO 12
412       414 CONTINUE
413           NTFL(L)=.FALSE.
414           IF(DAYFL(L)) GO TO 12
415           GO TO 500
416        12 CONTINUE
417     C
418     C               ***FILL SURFACE POTENTIAL TEMPERATURE ARRAY.
419     C
420     C               *SPECIFY VALUES AT ISR-4 AND ISR+2 THE NEXT DAY IF
421     C               *VALUES MISSING.  IF NO DAYTIME OR NIGHTIME VALUES
422     C               *SPECIFY VALUES AT ISS+2 (FIRST DAY).
423           IF(DAYFL(L)) THEN
424             DO 14 J=ISR-4,ISR
425             IF (APT(J,L).NE.999.) THEN
426               DO 418 K=ISR-4,J-1
427               APT(K,L)=APT(J,L)
428       418     CONTINUE
429               GO TO 419
430             ENDIF
431        14   CONTINUE
432     C               **(PROGRAM EXECUTION SHOULD NOT PASS THRU THIS POINT)
433       419   CONTINUE
434           ENDIF
```

```
C
      IF(NTFL(L)) THEN
        DO 430 J=ISR+26,ISR+20,-1
        IF(APT(J,L).NE.999.) THEN
          DO 429 K=ISR+26,J+1,-1
          APT(K,L)=APT(J,L)
  429     CONTINUE
          GO TO 431
        ENDIF
  430   CONTINUE
C             **(PROGRAM EXECUTION SHOULD NOT PASS THRU THIS POINT)
  431   CONTINUE
      ENDIF
C
      IF(.NOT.NTFL(L)) THEN
        IF(APT(ISS+2,L).EQ.999.) THEN
          IF(APT(ISS+1,L).EQ.999.) THEN
            APT(ISS+2,L)=APT(ISS,L)
            APT(ISS+1,L)=APT(ISS,L)
          ELSE
            APT(ISS+2,L)=APT(ISS+1,L)
          ENDIF
        ENDIF
      ENDIF
C
      IF(.NOT.DAYFL(L)) THEN
        IF(APT(ISS,L).EQ.999.) THEN
          IF(APT(ISS+1,L).EQ.999.) THEN
            APT(ISS,L)=APT(ISS+2,L)
            APT(ISS+1,L)=APT(ISS+2,L)
          ELSE
            APT(ISS,L)=APT(ISS+1,L)
          ENDIF
        ENDIF
      ENDIF
C
C             *LINEARLY INTERPOLATE TO GET MISSING VALUES.
      IC=0
      IF(DAYFL(L)) ISTART=ISR-4
      IF(.NOT.DAYFL(L)) ISTART=ISS
      IF(NTFL(L)) ISTOP=ISR+26
      IF(.NOT.NTFL(L)) ISTOP=ISS+2
      DO 16 J=ISTART,ISTOP
      IF (APT(J,L).NE.999.)  THEN
        IF (IC.NE.0)  THEN
          DT=(APT(J,L)-APT(J-IC+1,L))/FLOAT(IC+1)
          DO 15 K=J-IC,J-1
          APT(K,L)=APT(K-1,L)+DT
  15      CONTINUE
          IC=0
        ENDIF
      ELSE
        IC=IC+1
      ENDIF
  16  CONTINUE
C
C
C             ***FILL SURFACE TEMPERATURE ARRAY.
C
C             *SPECIFY VALUES AT ISR-4 AND ISR+2 THE NEXT DAY IF
C             *VALUES MISSING.  IF NO DAYTIME OR NIGHTIME VALUES
C             *SPECIFY VALUES AT ISS+2 (FIRST DAY).
```

```
497           IF(DAYFL(L)) THEN
498             DO 21 J=ISR-4,ISR
499             IF (ATMP(J,L).NE.999.) THEN
500               DO 20 K=ISR-4,J-1
501               ATMP(K,L)=ATMP(J,L)
502        20     CONTINUE
503               GO TO 22
504             ENDIF
505        21   CONTINUE
506     C              **(PROGRAM EXECUTION SHOULD NOT PASS THRU THIS POINT)
507        22   CONTINUE
508           ENDIF
509     C
510           IF(NTFL(L)) THEN
511             DO 24 J=ISR+26,ISR+20,-1
512             IF(ATMP(J,L).NE.999.) THEN
513               DO 23 K=ISR+26,J+1,-1
514               ATMP(K,L)=ATMP(J,L)
515        23     CONTINUE
516               GO TO 25
517             ENDIF
518        24   CONTINUE
519     C              **(PROGRAM EXECUTION SHOULD NOT PASS THRU THIS POINT)
520        25   CONTINUE
521           ENDIF
522     C
523           IF(.NOT.NTFL(L)) THEN
524             IF(ATMP(ISS+2,L).EQ.999.) THEN
525               IF(ATMP(ISS+1,L).EQ.999.) THEN
526                 ATMP(ISS+2,L)=ATMP(ISS,L)
527                 ATMP(ISS+1,L)=ATMP(ISS,L)
528               ELSE
529                 ATMP(ISS+2,L)=ATMP(ISS+1,L)
530               ENDIF
531             ENDIF
532           ENDIF
533     C
534           IF(.NOT.DAYFL(L)) THEN
535             IF(ATMP(ISS,L).EQ.999.) THEN
536               IF(ATMP(ISS+1,L).EQ.999.) THEN
537                 ATMP(ISS,L)=ATMP(ISS+2,L)
538                 ATMP(ISS+1,L)=ATMP(ISS+2,L)
539               ELSE
540                 ATMP(ISS,L)=ATMP(ISS+1,L)
541               ENDIF
542             ENDIF
543           ENDIF
544     C
545     C              *LINEARLY INTERPOLATE TO GET MISSING VALUES.
546           IC=0
547           IF(DAYFL(L)) ISTART=ISR-4
548           IF(.NOT.DAYFL(L)) ISTART=ISS
549           IF(NTFL(L)) ISTOP=ISR+26
550           IF(.NOT.NTFL(L)) ISTOP=ISS+2
551           DO 27 J=ISTART,ISTOP
552           IF (ATMP(J,L).NE.999.)  THEN
553             IF (IC.NE.0)  THEN
554               DT=(ATMP(J,L)-ATMP(J-IC+1,L))/FLOAT(IC+1)
555               DO 26 K=J-IC,J-1
556               ATMP(K,L)=ATMP(K-1,L)+DT
557        26     CONTINUE
558               IC=0
```

```
559             ENDIF
560           ELSE
561             IC=IC+1
562           ENDIF
563        27 CONTINUE
564     C
565     C               ***COMPUTE TEMPERATURE ADVECTION RATE.
566     C               ***COMPUTED AS THE POTENTIAL TEMPERATURE DIFFERENCE
567     C               ***BETWEEN THE MORNING AND AFTERNOON SOUNDINGS AT
568     C               ***3000 METERS ABOVE GROUND LEVEL.
569           IF(DAYFL(L)) THEN
570             HPMSL=ADHT+THTM(1,L)
571             DO 94 J=1,NOBM(L)
572             PPT(J)=PTM(J,L)
573             TH(J)=THTM(J,L)
574          94 CONTINUE
575             CALL TPROF (NOBM(L),TH,PPT,HPMSL,VALM)
576             HPMSL=ADHT+THTA(1,L)
577             DO 95 J=1,NOBA(L)
578             PPT(J)=PTA(J,L)
579             TH(J)=THTA(J,L)
580          95 CONTINUE
581             CALL TPROF (NOBA(L),TH,PPT,HPMSL,VALA)
582             TAR=(1./REAL(IHRA-IHRM))*(VALA-VALM)
583     C
584     C               ***FILL SURFACE POTENTIAL TEMPERATURE ARRAY FROM
585     C               ***SUNRISE TO 1 HOUR BEFORE SUNSET WITH RELATIVE TEMPERATUR
586             IC=0
587             DO 17 J=ISR,ISS-1
588             APT(J,L)=APT(J,L)-REAL(IC)*TAR
589             IC=IC+1
590          17 CONTINUE
591     C
592     C               ***CORRECT MORNING SOUNDING TO SURFACE TEMPERATURE.
593             NOBMM(L)=NOBM(L)
594             DO 18 J=1,NOBM(L)
595             THTMM(J,L)=THTM(J,L)
596             PTMM(J,L)=PTM(J,L)
597          18 CONTINUE
598             IF (PTM(1,L).NE.APT(ISR,L))  THEN
599               HMMSL=HMM+THTM(1,L)
600               IF (HMMSL.GE.THTM(NOBM(L),L))  THEN
601                 WRITE (6,34)
602                 STOP 'TEMP'
603               ENDIF
604               IF (HMM.LT.50.) HMM=50.
605               HMMSL=HMM+THTM(1,L)
606               DO 97 J=1,NOBM(L)
607               TH(J)=THTM(J,L)
608               PPT(J)=PTM(J,L)
609          97   CONTINUE
610               CALL TPROF (NOBM(L),TH,PPT,HMMSL,VALM)
611               PTMM(1,L)=APT(ISR,L)
612               PTMM(2,L)=VALM
613               THTMM(1,L)=THTM(1,L)
614               THTMM(2,L)=HMMSL
615               IC=2
616               DO 19 J=2,NOBM(L)
617               IF (THTM(J,L).LE.HMMSL) GO TO 19
618               IC=IC+1
619               THTMM(IC,L)=THTM(J,L)
620               PTMM(IC,L)=PTM(J,L)
```

```
621        19     CONTINUE
622               NOBMM(L)=IC
623             ENDIF
624           ENDIF
625     C
626     C              ***END OF LOOP ON STATIONS
627       500 CONTINUE
628     C
629     C
630           RETURN
631        33 FORMAT (2X,'ID OF SURFACE DATA DOES NOT CORRESPOND WITH ID REQUIRE
632          +D.')
633        34 FORMAT (2X,'MECHANICAL MIXING HEIGHT GREATER THAN HIGHEST OBSERVAT
634          +ION.')
635        51 FORMAT (2X,'NO STATIONS AVAILABLE TO COMPUTE POTENTIAL TEMP.',
636          +             ' LAPSE RATE AND BULK TEMP.')
637           END
```

```
001     C  SUBROUTINE: MET.TETRI
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO PROVIDE TERRAIN HEIGHT AND SLOPES
005     C  AT A POINT GIVEN A GRID OF TERRAIN HEIGHTS.  THE HEIGHT AT A POINT
006     C  IS THE NEAREST GRID ELEVATION TO THE POINT.  THE X-DIRECTION SLOPE
007     C  IS [THS(II+1,JJ)-THS(II-1,JJ)]/[2*DELTA ALPHA], AND THE Y-DIRECTION
008     C  SLOPE IS [THS(II,JJ+1)-THS(II,JJ-1)]/[2*DELTA BETA].
009     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
010     C  THE INPUTS ARE:
011     C    THRU ARGUMENT:
012     C      XG - ALPHA COORDINATE OF DESIRED POINT.
013     C      YG - BETA COORDINATE OF DESIRED POINT.
014     C    THRU COMMON BLOCK 'TETRI1':
015     C      NXCT = NUMBER OF GRID CELLS IN X-DIRECTION
016     C      NYCT = NUMBER OF GRID CELLS IN Y-DIRECTION
017     C      THS = ARRAY OF AVERAGE TERRAIN HEIGHTS IN GRID CELLS (M MSL)
018     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
019     C  THE OUTPUTS ARE:
020     C    THRU ARGUMENT:
021     C      TH - TERRAIN HEIGHT AT POINT (M MSL).
022     C      DZA - SLOPE IN ALPHA (X) DIRECTION ( M/[DELTA ALPHA] ).
023     C      DZB - SLOPE IN BETA (Y) DIRECTION ( M/[DELTA BETA] ).
024     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
025     C  CALLING SUBROUTINES:
026     C      BLPARS, FROUDE, MIXSTB, WINDS
027     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
028     C  CALLED SUBROUTINES:
029     C      (NONE)
030     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
031     C  COMMON BLOCKS:
032     C      TETRI1
033     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
034     C  PARAMETERS DEFINITIONS:
035     C      NPT = NUMBER OF GRID POINTS ACROSS DOMAIN IN TERRAIN FILE
036     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
037     C
038           SUBROUTINE TETRI (XG,YG,TH,DZA,DZB)
039     C
040           PARAMETER
041          +    NPT=50
042     C
043           COMMON /TETRI1/
044          +    ICST,          IX0T,          IY0T,          NXCT,
045          +    NYCT,          THS(NPT,NPT)
046     C
047           XC=REAL(NXCT)
048           YC=REAL(NYCT)
049           DX=1./XC
050           DY=1./YC
051     C
052     C              **CALCULATE ARRAY INDICIES.
053           IF(XG.LE.0.) THEN
054             II=1
055           ELSE
056             IF(XG.GE.1.) THEN
057               II=NXCT
058             ELSE
059               II=NINT(XC*XG-0.5)+1
060             ENDIF
061           ENDIF
062           IF(YG.LE.0.) THEN
```

```
063             JJ=1
064           ELSE
065             IF(YG.GE.1.) THEN
066               JJ=NYCT
067             ELSE
068               JJ=NINT(YC*YG-0.5)+1
069             ENDIF
070           ENDIF
071     C
072     C               **DETERMINE TERRAIN HEIGHT
073           TH=THS(II,JJ)
074     C
075     C               **COMPUTE SLOPE WITH RESPECT TO ALPHA
076           IF(II.EQ.1) THEN
077             DZA=(THS(2,JJ)-THS(1,JJ))/DX
078           ELSE
079             IF(II.EQ.NXCT) THEN
080               DZA=(THS(NXCT,JJ)-THS(NXCT-1,JJ))/DX
081             ELSE
082               DZA=(THS(II+1,JJ)-THS(II-1,JJ))/(2.*DX)
083             ENDIF
084           ENDIF
085     C
086     C               **COMPUTE SLOPE WITH RESPECT TO BETA
087           IF(JJ.EQ.1) THEN
088             DZB=(THS(II,2)-THS(II,1))/DY
089           ELSE
090             IF(JJ.EQ.NYCT) THEN
091               DZB=(THS(II,NYCT)-THS(II,NYCT-1))/DY
092             ELSE
093               DZB=(THS(II,JJ+1)-THS(II,JJ-1))/(2.*DY)
094             ENDIF
095           ENDIF
096     C
097           RETURN
098           END
```

```
001     C  SUBROUTINE: MET.THETA
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE SEA-LEVEL 'THETA' USED
005     C  IN THE DRY ADIABATIC APPROXIMATION FOR DETERMINING DENSITY
006     C  VARIATION WITH HEIGHT.  DENSITY VARIATION WITH HEIGHT IS
007     C  GIVEN BY THE RELATIONSHIP RHO(Z)=RHO(0)*(1-Z/HA)**2/5 WHERE,
008     C  RHO(0) IS 353.1/THETA (SEA-LEVEL DENSITY OF AIR), HA IS
009     C  102.4*THETA (HEIGHT OF ADIABATIC ATMOSPHERE), AND Z IS HEIGHT.
010     C  THIS RESULTS IN THE EQUATION:
011     C  RHO(Z)=(353.1/THETA)*(1-Z/(102.4*THETA))**2/5 WHICH IS
012     C  SOLVED FOR THETA USING NEWTON'S METHOD GIVEN RHO(Z) AND Z.
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  THE INPUTS ARE:
015     C       RHO = DENSITY OF AIR AT HEIGHT Z (KG/M**3)
016     C       Z = HEIGHT OF DENSITY MEASURE (M MSL)
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       TH = THETA [POT. TEMP. OF THE DRY ADIABATE (DEG K)]
020     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
021     C  CALLING SUBROUTINES:
022     C       AMPLIT, BLPARS
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLED SUBROUTINES:
025     C       (NONE)
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  COMMON BLOCKS:
028     C       (NONE)
029     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
030     C
031           SUBROUTINE THETA(RHO,Z,TH)
032     C
033           B=2./5.
034           A=(RHO/353.1)**B
035           C=Z/102.4
036           D=7./5.
037     C
038     C              ***USE NEWTON'S METHOD
039           IT=0
040           TH=300.
041         3 IT=IT+1
042             G=A*TH**D-TH+C
043             GP=D*A*TH**B-1.
044             R=G/GP
045             TH=TH-R
046             CK=ABS(R)
047             IF(CK.LE.1.) RETURN
048             IF(IT.GT.10) THEN
049               WRITE(6,4)
050               STOP 'THETA'
051             ENDIF
052           GO TO 3
053         4 FORMAT(2X,'THETA DID NOT CONVERGE IN 10 ITERATIONS.')
054     C
055           END
```

```
001     C  SUBROUTINE: MET.TPROF
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO DETERMINE A VALUE (E.G., POT. TEMP,
005     C  TEMP., OR PRESSURE) AT A DESIRED HEIGHT GIVEN AN UPPER AIR
006     C  SOUNDING OF THE APPROPRIATE PARAMETER VERSES HEIGHT.  A CONSTRAINT
007     C  IS THAT THE HEIGHT OF THE LAST OBSERVATION BE GREATER THAN THE
008     C  DESIRED HEIGHT.  IF THE SPECIFIED HEIGHT IS LESS THAN THE SURFACE
009     C  ELEVATION THEN THE DESIRED VALUE IS SET TO THE SURFACE VALUE.  IF
010     C  THE SPECIFIED HEIGHT IS GREATER THAN THE HEIGHEST OBSERVATION THEN
011     C  THE DESIRED VALUE IS SET TO THE VALUE OF THE HIGHEST OBSERVATION.
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE INPUTS ARE:
014     C       NOBS = NUMBER OF OBSERVATIONS IN SOUNDING .
015     C       THT = ARRAY OF OBSERVATION HEIGHTS (M MSL)  [OR M AGL].
016     C       PT = ARRAY OF OBSERVED VALUES (E.G. POT. TEMP. (DEG K),
017     C            TEMP. (DEG C), OR PRESSURE (MB)).
018     C       DH = HEIGHT OF DESIRED VALUE (M MSL)  [OR M AGL].
019     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
020     C  THE OUTPUTS ARE:
021     C       DPT = VALUE (E.G., POT. TEMP. (DEG K), TEMP. (DEG C), OR
022     C             PRESSURE (MB)) AT SPECIFIED HEIGHT.
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  CALLING SUBROUTINES:
025     C       BETA, COEFF, LAPSE, MIXSTB, TEMP, VALGAM, WARM
026     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
027     C  CALLED SUBROUTINES:
028     C       (NONE)
029     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
030     C  COMMON BLOCKS:
031     C       (NONE)
032     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
033     C  PARAMETER DEFINITIONS:
034     C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
035     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
036     C
037           SUBROUTINE TPROF (NOBS,THT,PT,DH,DPT)
038     C
039           PARAMETER
040          +      NOB=30
041     C
042           DIMENSION
043          +      PT(NOB),      THT(NOB)
044     C
045           IF (NOBS.LE.1)  THEN
046             WRITE (6,3)
047             STOP 'TPROF'
048           ENDIF
049     C
050     C              ***DETERMINE VALUE AT DESIGNATED HEIGHT.
051           IF (DH.LE.THT(1))  THEN
052             DPT=PT(1)
053             RETURN
054           ENDIF
055     C
056           IF (DH.GT.THT(NOBS))  THEN
057             DPT=PT(NOBS)
058             RETURN
059           ENDIF
060     C
061           DO 2 J=2,NOBS
062             IF (DH.LE.THT(J))  THEN
```

```
063               SLP=(PT(J)-PT(J-1))/(THT(J)-THT(J-1))
064               CPT=PT(J)-SLP*THT(J)
065               DPT=SLP*DH+CPT
066               RETURN
067             ENDIF
068         2 CONTINUE
069     C
070         3 FORMAT (2X,'NEED MORE THAN ONE OBSERVATION IN PROFILE.')
071           END
```

```
001     C  SUBROUTINE: MET.UPPER
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO READ IN UPPER AIR WIND AND TEMPERATURE
005     C  DATA FROM UNIT 3, DETERMINE WINDS ON SURFACES AND WRITE OUT THE
006     C  WIND DATA TO A DIRECT ACCESS DISK FILE ON UNIT 8.  THE
007     C  TEMPERATURE DATA IS WRITTEN OUT TO A DIRECT ACCESS DISK FILE ON
008     C  UNIT 9.
009     C    THE FOLLOWING QUALIFIERS APPLY TO THE USE OF THIS SUBROUTINE:
010     C      1. INPUT SOUNDINGS MUST BE GROUPED BY LOCATION AND WITHIN EACH
011     C      GROUPING THE SOUNDINGS MUST BE ORDERED BY TIME.
012     C      2. FOR A GIVEN SOUNDING THE WIND OBSERVATIONS MUST BE AFTER THE
013     C      TEMPERATURE OBSERVATIONS.
014     C      3. EVERY TEMPERATURE SOUNDING MUST HAVE THE SURFACE PERSSURE
015     C      (MB) AND ELEVATION (M MSL).  IN THE SOUNDING, HEIGHT OR PRESSURE
016     C      OBSERVATIONS CAN BE MISSING.
017     C      4. WIND SOUNDINGS MUST CONTAIN HEIGHTS (M MSL) FOR  ALL
018     C      OBSERVATIONS.  IF ANY HEIGHTS ARE MISSING THEN THE WIND
019     C      SOUNDING MUST HAVE A 1-TO-1 CORRESPONDENCE TO THE TEMP SOUNDING.
020     C      5. THE HEIGHT OF THE FIRST OBSERVATION MUST BE GREATER THAN OR
021     C      EQUAL TO THE SPECIFIED STATION ELEVATION.
022     C      6. THE COORDINATES OF THE STATION LOCATION CAN BE IN EITHER UTM
023     C      COORDINATES (ZONE 12 OR 13), LATITUDE-LONGITUDE, OR GRAMA
024     C      COORDINATES.
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  THE INPUTS ARE:
027     C    THRU ARGUMENT:
028     C      NSTA = NUMBER OF UPPER AIR STATIONS.
029     C      NGAMA = NUMBER OF GAMMA SURFACES.
030     C      HTGAMA = ARRAY OF HEIGHTS OF GAMMA SURFACES (FIRST LEVEL SHOULD
031     C               BE HEIGHT OF SURFACE MEASUREMENTS).
032     C      DS = SEED NUMBER FOR RANDOM WIND DIRECTION ADJUSTMENT
033     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL), 7(MOUNTAIN),
034     C                              8(PACIFIC).
035     C      TZNAM = BASE TIME ZONE NAME
036     C      DETAIL = LOGICAL VARIABLE IF .TRUE. MEANS WRITE OUT RUN DATA TO
037     C               UNIT LUO.
038     C      LUO = LOGICAL UNIT TO WHICH PRINT FILES ARE DIRECTED
039     C    THRU COMMON BLOCK 'MAIN2':
040     C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
041     C            1 - 'SMALL H' EQUAL TO CONSTANT
042     C            2 - 'BIG H' EQUAL TO CONSTANT
043     C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
044     C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
045     C            (SMALL H = BIG H - TERRAIN HEIGHT)
046     C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
047     C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
048     C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
049     C           TO HH)
050     C      IX0 = X-COORD ORIGIN OF DOMAIN (KM)
051     C      IY0 = Y-COORD ORIGIN OF DOMAIN (KM)
052     C      SLX = X-DIRECTION DOMAIN SIZE (KM)
053     C      SLY = Y-DIRECTION DOMAIN SIZE (KM)
054     C    THRU LOGICAL UNIT 3:
055     C      THE INPUT DATA IS IN TWO MAJOR GROUPINGS.  FIRST IS A SECTION
056     C      CONTAINING A RECORD FOR EACH STATION IN WHICH THE FOLLOWING
057     C      PARAMETERS ARE DEFINED ON EACH RECORD:
058     C      IDU = STATION IDENTIFICATION NUMBER.
059     C      NAME = STATION NAME (A10).
060     C      COX = X COORDINATE OF STATION.
061     C      COY = Y COORDINATE OF STATION.
062     C      EL = STATION ELEVATION (M. MSL).
```

```
063     C       ICODE = CODE SPECIFYING COORDINATE SYSTEM
064     C               1 = LATITUDE-LONGITUDE
065     C               2 = UTM ZONE 12
066     C               3 = UTM ZONE 13
067     C               4 = GRAMA
068     C       NSDT = NUMBER OF TEMPERATURE SOUNDINGS.
069     C       NSDW = NUMBER OF WIND SOUNDINGS
070     C       ITZU = BASE TIME ZONE FOR STATION = 5(EASTERN), 6(CENTRAL),
071     C                                           7(MOUNTAIN), 8(PACIFIC).
072     C
073     C       A SAMPLE RECORD FOR GRAND JUNCTION IS:
074     C       12345678901234567890123456789012345678901234567890123456789012
075     C            1 GRND JUNCT    39.117   108.533  1474.0   1   5   5   7
076     C
077     C       FOLLOWING THE PARAMETER SPECIFICATION RECORDS ARE THE SOUNDINGS.
078     C       THE FIRST RECORD OF THE SOUNDING IS AN IDENTIFICATION RECORD.
079     C       IT IDENTIFIES THE FOLLOWING PARAMETERS:
080     C       IID = STATION IDENTIFICATION NUMBER.
081     C       ITIME = TIME OF SOUNDING IN  "MO DA YR HR" (INTEGER 8).
082     C       NOBT = NUMBER OF TEMPERATURE OBSERVATIONS.
083     C       NOBW = NUMBER OF WIND OBSERVATIONS.
084     C       ITFLAG = SPECIFIES TMP. OBS. HEIGHTS INPUT
085     C              = 0 [ALL PRESSURE AND HEIGHT OBSERVATIONS GIVEN]
086     C              = 1 [ALL PRESS OBS (MB) GIVEN, NO HEIGHTS GIVEN (M MSL)]
087     C              = 2 [ALL HEIGHTS GIVEN (M MSL), NO PRESS OBS GIVEN (MB)]
088     C              = 3 [SOME PRESS (MB) OR HEIGHT (M MSL) OBS MISSING]
089     C       IWFLAG = SPECIFIES WIND OBS. HEIGHTS INPUT.
090     C              = 1 [ALL OBS HEIGHTS (M MSL) GIVEN]
091     C              = 2 [MISSING OBS HEIGHTS (M MSL), THEN WIND OBSERVATIONS
092     C                  MUST CORRESPOND 1-TO-1 WITH THE TEMP OBS. THE WIND
093     C                  OBS HEIGHTS (M MSL) ARE TAKEN FROM THE TEMP SND]
094     C       IMOIS = INDICATES WHETHER DEW POINT OR RELATIVE HUMIDITY INPUT.
095     C             = 1 [DEW POINT TEMPERATURE (DEG C)]
096     C             = 2 [RELATIVE HUMIDITY (%)]
097     C       IWU = INDICATES UNITS OF WIND DATA INPUT.
098     C           = 1 [WINDS INPUT IN (KNOTS)]
099     C           = 2 [WINDS INPUT IN (M/S)]
100     C       ITZO = TIME ZONE OF DATA = 0(ZULU), 5(EASTERN), 6(CENTRAL),
101     C                                           7(MOUNTAIN), 8(PACIFIC).
102     C
103     C       A SAMPLE RECORD FOR GRAND JUNCTIONS IS:
104     C       12345678901234567890123456789012345
105     C            1  08128017  50  50 3 1 1 1 0      FORMAT(I6,I10,2I4,5I2)
106     C
107     C       FOLLOWING THIS RECORD COMES THE TEMPERATURE OBSERVATIONS AND
108     C       THEN THE WIND OBSERVATIONS.
109     C       EACH TEMPERATURE OBSERVATION CONTAINS:
110     C       THTP = HEIGHT OF OBS IN MBARS.  (0 MEANS MISSING VALUE)
111     C       THTM = HEIGHT OF OBS IN M MSL.  (0 MEANS MISSING VALUE)
112     C       TMP = TEMPERATURE (DEG C).
113     C       RH = DEW POINT (DEG. C) OR RELATIVE HUMIDITY (%).
114     C       EACH WIND OBSERVATION CONTAINS:
115     C       WHTM = HEIGHT OF OBS IN M MSL.  (0 MEANS MISSING VALUE)
116     C       WD = WIND DIRECTION (DEG.)
117     C       WS = WIND SPEED (KNOTS) OR (M/S).
118     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
119     C  THE OUTPUTS ARE:
120     C    THRU COMMON BLOCK 'UPPER1':
121     C       COX = ARRAY OF X-COORD FOR STATIONS (KM)
122     C       COY = ARRAY OF Y-COORD FOR STATIONS (KM)
123     C       EL = ARRAY OF STATION ELEVATIONS (M MSL)
124     C       IDU = ARRAY OF STATION ID NUMBERS
```

```
C       NAME = ARRAY OF STATION NAMES
C       NSDT = ARRAY OF NUMBER OF TEMP SNDGS FOR STATIONS
C       NSDW = ARRAY OF NUMBER OF WIND SNDGS FOR STATIONS
C       XLAT = ARRAY OF LATITUDE OF STATIONS (DEG)
C       YLONG = ARRAY OF LONGITUDE OF STATIONS (DEG)
C       IUSRTW = ARRAY OF STARTING RECORD NUMBER OF WIND SNDGS FOR
C                STATIONS ON DIRECT ACCESS DISK FILE ON UNIT 8
C       IUSRTT = ARRAY OF STARTING RECORD NUMBER OF TEMP SNDGS FOR
C                STATIONS ON DIRECT ACCESS DISK FILE ON UNIT 9
C       IUENDW = ARRAY OF ENDING RECORD NUMBER OF WIND SNDGS
C       IUENDT = ARRAY OF ENDING RECORD NUMBER OF TEMP SNDGS
C       ITZU = BASE TIME ZONE FOR STATION = 5(EASTERN), 6(CENTRAL),
C                                           7(MOUNTAIN), 8(PACIFIC).
C     THRU LOGICAL UNIT 8:
C       WINDS ON EACH SURFACE ARE WRITTEN OUT TO AN UNFORMATTED DIRECT
C       ACCESS DISK FILE ON UNIT 8.  EACH RECORD REPRESENTS ONE SOUNDING.
C       EACH RECORD CONTAINS THE FOLLOWING:
C       IDU = STATION IDENTIFICATION NUMBER.
C       IJUL = JULIAN DATE IN BASE TIME ZONE
C       IHR = HOUR OF SOUNDING IN BASE TIME ZONE.
C       WX = X COMPONENT OF WIND (M/S).
C       WY = Y COMPONENT OF WIND (M/S).
C       (WX AND WY ARE REPEATED FOR EACH SURFACE FROM LOWEST UP)
C     THRU LOGICAL UNIT 9:
C       OBSERVED TEMPERATURE AND DEW POINT ARE WRITTEN OUT TO
C       AN UNFORMATTED DIRECT ACCESS DISK FILE ON UNIT 9.  EACH RECORD
C       REPRESENTS ONE SOUNDING.  EACH RECORD CONTAINS THE FOLLOWING:
C       IDU = STATION IDENTIFICATION NUMBER.
C       IJUL = JULIAN DATE IN BASE TIME ZONE.
C       IHR = HOUR OF SOUNDING IN BASE TIME ZONE.
C       HMMD = HEIGHT OF MECHANICAL MIXING DEPTH (M).
C       NOBT = NUMBER OF OBSERVATIONS.
C       THTP = HEIGHT OF OBS IN MBARS.
C       THTM = HEIGHT OF OBS (M MSL).
C       TMP = TEMPERATURE (DEG. C).
C       PT = POTENTIAL TEMPERATURE (DEG K)
C       (THTP, THTM, PT, AND TMP ARE REPEATED NOBT TIMES FROM LOWEST
C       HEIGHT UP).
C
C       THE SOUNDINGS ARE GROUPED BY LOCATION AND WITHIN EACH GROUPING
C       ORDERED BY TIME FROM PROCESSING START TO FINISH.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  FILES READ FROM:
C       LOGICAL UNIT 3 = SEQUENTIAL FORMATTED FILE OF UPPER AIR DATA
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  FILES WRITTEN TO:
C       LOGICAL UNIT  8 = DIRECT ACCESS UNFORMATTED DISK FILE OF UPPER
C                         AIR WIND DATA ON EACH 'GAMMA' SURFACE
C       LOGICAL UNIT  9 = DIRECT ACCESS UNFORMATTED DISK FILE OF UPPER
C                         AIR TEMPERATURE DATA
C       LOGICAL UNIT LUO = IF DETAIL IS .TRUE. THEN RUN INFO IS OUTPUT
C       LOGICAL UNIT  6 = ERROR MESSAGES
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C       AAMAIN
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C       FILLMB, FILLME, GRID, HGTS, JULIAN, LAT, LAY, MBARS, RDM,
C       SMALLH,  TPROF,  UTM, DEW, TETRI
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C       MAIN2, UPPER1
```

```
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C       NUA = NUMBER OF UPPER AIR STATIONS
C       NOBMX = MAX NUMBER OF OBS ALLOWED IN TEMP OR WIND SOUNDING
C       NOB = TWO TIMES NOBMX (ALLOWS ROOM IN ARRAYS TO PROCESS SNDGS)
C       NGA = NUMBER OF 'GAMMA' SURFACES
C       LREW = 2*NGA+3, [LENGTH OF RECORD IN WIND DISK FILE (WORDS)]
C       LRET = 4*NOBMX+5
C            = [LENGTH OF RECORD IN TEMPERATURE DISK FILE (WORDS)]
C       NREW = TOTAL # OF RECORDS IN WIND DISK FILE.
C       NRET = TOTAL # OF RECORDS IN TEMPERATURE DISK FILE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE UPPER (NSTA,NGAMA,HTGAMA,DS,IBTZ,TZNAM,DETAIL,LUO)
C
      PARAMETER
     +    LRET=65,        LREW=21,        NGA=9,          NOB=30,
     +    NRET=600,       NREW=600,       NUA=10,         NOBMX=15
C
      LOGICAL
     +    DETAIL
C
      DIMENSION
     +    DP(NOB),        HTGAMA(NGA),    HTGAMU(NGA,NUA),PT(NOB),
     +    THTM(NOB),      THTP(NOB),      TMP(NOB),       TOPL(NGA),
     +    WHTM(NOB),      WIN(NOB),       WOUT(NGA),      WX(NOB),
     +    WY(NOB),        WXG(NGA),       WYG(NGA)
C
      CHARACTER*10
     +    NAME,           TZNAM
C
      DOUBLE PRECISION
     +    DS
C
      COMMON /MAIN2/
     +    HH,             IHH,            IX0,            IY0,
     +    RXY,            SLX,            SLY
C
      COMMON /UPPER1/
     +    COX(NUA),       COY(NUA),       EL(NUA),        IDU(NUA),
     +    NAME(NUA),      NSDT(NUA),      NSDW(NUA),      XLAT(NUA),
     +    YLONG(NUA),     IUSRTW(NUA),    IUSRTT(NUA),    IUENDW(NUA),
     +    IUENDT(NUA),    ITZU(NUA)
C
C
      DTOR=3.14159/180.
      X0=REAL(IX0)
      Y0=REAL(IY0)
C
C             ***FIRST READ IN STATION INFORMATION.
      DO 54 J=1,NSTA
        READ (3,22) IDU(J),NAME(J),XU,YU,EL(J),ICODE,NSDT(J),
     +              NSDW(J),ITZU(J)
        GO TO (2,1,1,3), ICODE
    1   IZONE=ICODE
        CALL UTM (IZONE,XU,YU,COX(J),COY(J))
        CALL GRID (COX(J),COY(J),XLAT(J),YLONG(J))
        GO TO 4
    2   XLAT(J)=XU
        YLONG(J)=YU
        CALL LAT (XLAT(J),YLONG(J),COX(J),COY(J))
        GO TO 4
```

```
249         3   COX(J)=XU
250             COY(J)=YU
251             CALL GRID (COX(J),COY(J),XLAT(J),YLONG(J))
252         4 CONTINUE
253       C               **COMPUTE SURFACE HEIGHTS.
254           AL=(COX(J)-X0)/SLX
255           BE=(COY(J)-Y0)/SLY
256           IF(AL.GT.1..OR.AL.LT.0..OR.BE.GT.1..OR.BE.LT.0.) THEN
257             CALL SMALLH (IHH,HH,EL(J),AL,BE,SH)
258           ELSE
259             CALL TETRI (AL,BE,TH,DZA,DZB)
260             CALL SMALLH (IHH,HH,TH,AL,BE,SH)
261           ENDIF
262           DO 14 LM=1,NGAMA
263             HTGAMU(LM,J)=HTGAMA(LM)*SH+EL(J)
264        14 CONTINUE
265        54 CONTINUE
266       C****************WRITE OUT RUN INFO TO UNIT LUO************************
267           WRITE(LUO,40) TZNAM
268           WRITE(LUO,41)
269           DO 60 J=1,NSTA
270             WRITE(LUO,42) NAME(J),IDU(J),COX(J),COY(J),XLAT(J),YLONG(J),
271            +              EL(J),NSDT(J),NSDW(J)
272        60 CONTINUE
273           WRITE(LUO,43) (HTGAMA(NG),NG=1,NGAMA)
274           DO 61 J=1,NSTA
275             WRITE(LUO,44) NAME(J),(HTGAMU(NG,J),NG=1,NGAMA)
276        61 CONTINUE
277       C*******************************************************************************
278       C
279       C               ***DEFINE DIRECT ACCESS FILES FOR WIND AND TEMPERATURE
280       C               ***DATA.
281           DEFINE FILE 8 (NREW,LREW,U,IIW)
282           DEFINE FILE 9 (NRET,LRET,U,IIT)
283           IIW=1
284           IIT=1
285       C
286       C               ***PROCESS UPPER AIR DATA.
287           DO 18 JJ=1,NSTA
288       C****************WRITE OUT RUN INFO TO UNIT LUO************************
289           IF(DETAIL) THEN
290             WRITE(LUO,45) NAME(JJ)
291           ENDIF
292       C*******************************************************************************
293           IF (NSDW(JJ).GE.NSDT(JJ)) NSD=NSDW(JJ)
294           IF (NSDW(JJ).LT.NSDT(JJ)) NSD=NSDT(JJ)
295           DO 17 J=1,NSD
296       C
297       C               ***READ IN IDENTIFICATION RECORD FOR SOUNDING.
298           READ (3,23) IID,ITIME,NOBT,NOBW,ITFLAG,IWFLAG,IMOIS,IWU,ITZO
299           IF (IID.NE.IDU(JJ))  THEN
300             WRITE (6,24)
301             STOP 'UPPER'
302           ENDIF
303           CALL JULIAN (ITIME,ITZO,IBTZ,IHR,IJUL)
304       C
305       C               ***PROCESS TEMPERATURE SOUNDING.
306           HTOPT=-999.
307           IF (NOBT.EQ.0) GO TO 8
308           DO 5 K=1,NOBT
309             READ(3,25) THTMI,THTPI,TMPI,RH
310             IF(K.LE.NOBMX) THEN
```

```
311             THTM(K)=THTMI
312             THTP(K)=THTPI
313             TMP(K)=TMPI
314             DP(K)=RH
315             IF(IMOIS.EQ.2) CALL DEW(TMP(K),RH,DP(K))
316           ENDIP
317         5 CONTINUE
318           IF(NOBT.GT.NOBMX) NOBT=NOBMX
319           IF( ITFLAG.EQ.0) GO TO 6
320           IF (ITFLAG.EQ.1) CALL HGTS (NOBT,THTP,THTM,TMP,DP,EL(JJ))
321           IF (ITFLAG.EQ.2) CALL MBARS (NOBT,THTP,THTM,TMP,DP)
322           IF (ITFLAG.EQ.3)  THEN
323             DO 6 K=2,NOBT
324               IF (THTP(K).EQ.0.) CALL FILLMB (THTM(K),THTM(K-1),THTP(K),
325            +                                  THTP(K-1),TMP(K),TMP(K-1),DP(K),DP(K-1))
326               IF (THTM(K).EQ.0.) CALL FILLME (THTM(K),THTM(K-1),THTP(K),
327            +                                  THTP(K-1),TMP(K),TMP(K-1),DP(K),DP(K-1))
328         6   CONTINUE
329           ENDIF
330           DO 7 K=1,NOBT
331             PT(K)=(TMP(K)+273.15)*(1000./THTP(K))**.286
332         7 CONTINUE
333           HTOPT=THTM(NOBT)
334         8 CONTINUE
335     C
336     C             ***PROCESS WIND SOUNDING.
337           HTOPW=-999.
338           HMMD=0.
339           IF (NOBW.EQ.0) GO TO 16
340           DO 9 K=1,NOBW
341             READ(3,28) WHTMI,WD,WS
342             IF(NOBW.LE.NOBMX) THEN
343               WHTM(K)=WHTMI
344     C             **CONVERT WIND SPEED FROM KNOTS TO M/S IF INPUT IN KNOTS.
345               IF(IWU.EQ.1) WS=WS*0.5148
346     C             **RANDOMIZE WIND DIRECTION IF INPUT ON WHOLE 10 DEGREES.
347     C             **THE CORRECTION IS FROM -4 TO 5 DEGREES.
348               CKW=REAL(INT(WD/10.))*10.
349               IF (WD.EQ.CKW)  THEN
350                 CALL RDM (DS,OUT)
351                 WD=WD+OUT
352                 IF (WD.GT.360.) WD=WD-360.
353                 IF (WD.LT.0.) WD=360.+WD
354               ENDIF
355     C             ***RESOLVE WIND INTO COMPONENTS
356               THET=(270.-WD)*DTOR
357               WX(K)=WS*COS(THET)
358               WY(K)=WS*SIN(THET)
359               WIN(K)=WS
360             ENDIF
361         9 CONTINUE
362           IF(NOBW.GT.NOBMX) NOBW=NOBMX
363     C             ***SPECIFY WIND OBSERVATION HEIGHTS FROM TEMP OBS
364     C             ***HEIGHTS IF WIND OBS HEIGHTS MISSING.
365           IF (IWFLAG.EQ.2) THEN
366             IF(NOBW.NE.NOBT) THEN
367               WRITE (6,26)
368               STOP 'UPPER'
369             ENDIF
370             DO 11 KK=1,NOBW
371               WHTM(KK)=THTM(KK)
372        11   CONTINUE
```

```
373           ENDIF
374     C               ***DETERMINE WINDS AT EACH SURFACE.
375           IF (HTGAMU(NGAMA,JJ).GT.WHTM(NOBW))  THEN
376             NOBW=NOBW+1
377             WHTM(NOBW)=HTGAMU(NGAMA,JJ)
378             WX(NOBW)=WX(NOBW-1)
379             WY(NOBW)=WY(NOBW-1)
380           ENDIF
381            IF(WHTM(1).LT.HTGAMU(1,JJ)) WHTM(1)=HTGAMU(1,JJ)
382            DO 12 K=1,NGAMA
383             CALL TPROF(NOBW,WHTM,WX,HTGAMU(K,JJ),WXG(K))
384             CALL TPROF(NOBW,WHTM,WY,HTGAMU(K,JJ),WYG(K))
385        12 CONTINUE
386     C
387     C               ***COMPUTE HEIGHT OF MECH MIXING HEIGHT USING METHOD
388     C               ***GIVEN IN MESOPUFF.  MECHANICAL MIXING HEIGHT = 53 *
389     C               ***(FREE STREAM WIND SPEED).  FREE STREAM WIND SPEED IS
390     C               ***BEING COMPUTED AS THE AVERAGE SPEED BETWEEN THE TOP
391     C               ***TWO GAMMMA SURFACES.
392           NLAYS=NGAMA-1
393           DO 15 LM=1,NLAYS
394             TOPL(LM)=HTGAMU(LM+1,JJ)
395        15 CONTINUE
396           CALL LAY (NOBW,WHTM,WIN,WOUT,NLAYS,TOPL)
397           HMMD=WOUT(NLAYS)*53.
398           HTOPW=WHTM(NOBW)
399        16 CONTINUE
400     C
401     C               ***WRITE OUT WIND AND TEMPERATURE SOUNDINGS.
402           IF (NOBW.NE.0) WRITE (8'IIW)IDU(JJ),IJUL,IHR,(WXG(K),WYG(K),
403          +                          K=1,NGAMA)
404           IF (NOBT.NE.0) WRITE (9'IIT)IDU(JJ),IJUL,IHR,HMMD,NOBT,(THTP(K),
405          +                          THTM(K),TMP(K),PT(K),K=1,NOBT)
406     C****************WRITE OUT RUN INFO TO UNIT LUO************************
407           IF(DETAIL) THEN
408             WRITE(LUO,46) IJUL,IHR,NOBT,HTOPT,NOBW,HTOPW
409           ENDIF
410     C*******************************************************************************
411        17 CONTINUE
412        18 CONTINUE
413     C
414     C               ***COMPUTE STARTING AND ENDING RECORD FOR STATION.
415           IUSRTW(1)=1
416           IUSRTT(1)=1
417           IUENDW(1)=IUSRTW(1)+NSDW(1)-1
418           IUENDT(1)=IUSRTT(1)+NSDT(1)-1
419           DO 30 J=2,NSTA
420             IUSRTW(J)=IUSRTW(J-1)+NSDW(J-1)
421             IUSRTT(J)=IUSRTT(J-1)+NSDT(J-1)
422             IUENDW(J)=IUSRTW(J)+NSDW(J)-1
423             IUENDT(J)=IUSRTT(J)+NSDT(J)-1
424        30 CONTINUE
425     C
426     C
427           RETURN
428        24 FORMAT (5X,'STATION TO BE PROCESSED NOT SAME AS SOUNDING READ.')
429        22 FORMAT(I6,1X,A10,2F10.3,F8.1,4I4)
430        23 FORMAT(I6,I10,2I4,5I2)
431        25 FORMAT(2F7.1,2F6.1)
432        28 FORMAT(F7.1,F6.1,F5.1)
433        26 FORMAT (5X,'HEIGHTS FROM WIND AND TMP SOUNDINGS NOT EQUAL.')
434        40 FORMAT(1H1,///,5X,'TRACE OF SUBROUTINE UPPER EXECUTION - UPPER',
```

```
435         +' READS IN ALL UPPER AIR DATA FROM THE INPUT FILE AND WRITES',
436         +/,5X,'OUT DIRECT ACCESS DISK FILES FOR USE BY OTHER PROGRAM',
437         +' COMPONENTS.  THE BASE TIME ZONE IS',A10,'.' )
438       41 FORMAT(1H0,/,5X,'SUMMARY OF STATIONS PROCESSED:',/,10X,'NAME',
439         +6X,'ID NO',2X,'X (KM)',2X,'Y (KM)',2X,'LAT (DEG)',2X,
440         +'LONG (DEG)',2X,'ELEV (M MSL)',2X,'NO TMP SND',2X,'NO WND SND')
441       42 FORMAT(1H ,6X,A10,2X,I6,2X,F6.1,2X,F6.1,3X,F7.3,5X,F7.3,6X,F6.1,
442         +8X,I3,9X,I3)
443       43 FORMAT(1H0,/,5X,'THE WINDS FOR EACH STATION ARE OUTPUT AT THE',
444         +' FOLLOWING HEIGHTS (M MSL) CORRESPONDING TO THE IDENTIFIED',/,
445         +5X,'"GAMMA SURFACES" :',/,10X,'NAME',4X,11(2X,'"',F6.4,'"'))
446       44 FORMAT(1H ,6X,A10,11(4X,F6.1))
447       45 FORMAT(1H0,/,5X,'THE SOUNDINGS PROCESSED FOR STATION "',A10,
448         +'" ARE:',/,8X,'DAY',2X,'HOUR',2X,'NO TMP OBS',2X,'HT TOP TMP OBS'
449         +,2X,'NO WND OBS',2X,'HT TOP WND OBS')
450       46 FORMAT(1H ,7X,I3,3X,I2,7X,I3,9X,F6.1,10X,I3,9X,F6.1)
451          END
```

```
001     C  SUBROUTINE: MET.USTAR
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE SURFACE FRICTION
005     C  VELOCITY AFTER THE APPROACH GIVEN IN MESOPUFF II (MARCH 1984).
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C       TSURF = SURFACE TEMPERATURE (DEG. K)
009     C       WS = WIND SPEED AT REFERENCE HEIGHT (M/S)
010     C       ZMS = REFERENCE HEIGHT OF WIND SPEED MEASUREMENT (M)
011     C       Z0 = SURFACE ROUGHNESS (M)
012     C       WT = W'THETA' ((M-DEGK)/S)
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C  THE OUTPUTS ARE:
015     C       USTR = SURFACE FRICTION VELOCITY (M/S)
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  CONSTANT DEFINITIONS:
018     C       VK = 0.4  [VON KARMAN CONSTANT]
019     C       GRAV = 9.81  [ACCELERATION DUE TO GRAVITY (M/S**2)]
020     C       GAMMA = 4.7
021     C       CONSTA = 1100.
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  CALLING SUBROUTINES:
024     C       BLPARS
025     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
026     C  CALLED SUBROUTINES:
027     C       (NONE)
028     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
029     C  COMMON BLOCKS:
030     C       (NONE)
031     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
032     C
033           SUBROUTINE USTAR(TSURF,WS,ZMS,Z0,WT,USTR)
034     C
035     C                 ***CONSTANTS
036           VK=0.4
037           GRAV=9.81
038           GAMMA=4.7
039           CONSTA=1100.
040     C
041     C                 ***IF CALM, USTAR IS ZERO
042           IF(WS.EQ.0.)GO TO 50
043           IF(WT.LE.0.0)GO TO 100
044     C                 ***BOUNDARY LAYER CONVECTIVE WHEN W'THETA' > 0
045     C                 ***ZM IS ZMS - D (D=ZERO PLANE DISPL. CONST. = 4.*Z0)
046           ZM=ZMS-4.*Z0
047           R1=Z0/ZM
048           ALNR1=ALOG(R1)
049           A=0.107
050           IF(R1.LE.0.01)A=0.128+0.005*ALNR1
051           B=1.95+32.6*R1**0.45
052           UTW=VK*WS/ALOG(ZM/Z0)
053           QTW=TSURF*UTW**3/(VK*GRAV*ZM)
054           USTR=UTW*(1.+A*ALOG(1.+B*WT/QTW))
055           GO TO 500
056        50 CONTINUE
057           USTR=0.0
058           GO TO 500
059       100 CONTINUE
060     C                 ***STABLE BOUNDARY LAYER
061           ZM=ZMS-4.*Z0
062           CDN=VK/ALOG(ZM/Z0)
```

```
063           U02=GAMMA*ZM/(VK*CONSTA)
064           C=1.-4.*U02/(CDN*WS**2)
065           C=AMAX1(C,0.0)
066           USTR=CDN*WS*(1.+SQRT(C))/2.
067       500 CONTINUE
068       600 CONTINUE
069     C
070           RETURN
071           END
```

```
C  SUBROUTINE: MET.UTM
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO CONVERT FROM UNIVERSAL TRANSVERSE
C  MERCATOR (UTM) COORDINATES IN KILOMETERS TO GRAMA COORDINATES IN
C  KILOMETERS.  THE GRAMA GRID IS CONTAINED WITHIN UTM GRID ZONES
C  12 AND 13.  EQUATIONS FOR A TRANSVERSE MERCATOR PROJECTION
C  ASSUMING THE EARTH IS SHAPED AS A SPHERE ARE USED.  THIS IS AN
C  APPROXIMATION TO THE UTM GRID IN WHICH THE TRANSFORMATION
C  EQUATIONS ARE BASED ON THE EARTH SHAPED AS AN ELLIPSOID.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C       IZONE = 2  (UTM GRID ZONE 12) OR,
C               3  (UTM GRID ZONE 13).
C       UTME = UTM EASTING COORDINATE (KM)
C       UTMN = UTM NORTHING COORDINATE (KM)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C       COX = X-COORDINATE IN GRAMA GRID SYSTEM (KM)
C       COY = Y-COORDINATE IN GRAMA GRID SYSTEM (KM)
C  CONSTANT DEFINITIONS:
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C       RAD = 6367.3858 [MEAN RADIUS OF EARTH (KM)]
C       C = .9996 [PROJECTION SCALING FACTOR]
C       CM = 111. [CENTRAL MERIDIAN OF UTM ZONE 13 (DEG)]
C          = 105. [CENTRAL MERIDIAN OF UTM ZONE 12 (DEG)]
C       OX = 499.472 [ADJ. ORIG. OF ZONE 13 WEST OF CM (KM)]
C          = 500.264 [ADJ. ORIG. OF ZONE 12 WEST OF CM (KM)]
C       OY = 15.943 [ADJ. ORIG. OF ZONE 13 NORTH OF EQTR. (KM)]
C          = 15.947 [ADJ. ORIG. OF ZONE 12 NORTH OF EQTR. (KM)]
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C       AMPLIT, COEFF, SRFC, UPPER
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C       LAT
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C       (NONE)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE UTM (IZONE,UTME,UTMN,COX,COY)
C
      DOUBLE PRECISION B,XPRIM,YPRIM,A,D
C
C             ***CONSTANTS
      CO1=180./3.141592654
      CO2=3.141592654/2.
C
      IF (IZONE.EQ.2) GO TO 1
C
C             ***DEFINE CONSTANT PARAMETERS FOR ZONE 13.
      R=6367.3858
      C=.9996
      CM=105.
      X0=499.472
      Y0=15.943
      GO TO 2
C
C             ***DEFINE CONSTANT PARAMETERS FOR ZONE 12.
    1 CONTINUE
      R=6367.3858
```

```
063           C=.9996
064           CM=111.
065           X0=500.264
066           Y0=15.947
067     C
068     C               ***COMPUTE LATITUDE-LONGITUDE IN DEGREES.
069         2 CONTINUE
070           B=EXP(2.*(UTME-X0)/(C*R))
071           XPRIM=(B-1.)/(B+1.)
072           YPRIM=TAN(CO2-(UTMN+Y0)/(C*R))
073           A=DSQRT((1.-XPRIM**2)/(1.+YPRIM**2))
074           D=DACOS(YPRIM*A/DSQRT(1.-A**2))*CO1
075           XLAT=DASIN(A)*CO1
076           IF(XPRIM.LT.0.) THEN
077             YLONG=CM+D
078           ELSE
079             YLONG=CM-D
080           ENDIF
081     C
082     C               ***CONVERT LATITUDE-LONGITUDE TO GRAMA COORDINATES.
083           CALL LAT (XLAT,YLONG,COX,COY)
084     C
085           RETURN
086           END
```

```
C  SUBROUTINE: VALGAM.MET
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE PROVIDES AN ESTIMATE OF THE DENSITY, PRESSURE,
C  AND POTENTIAL TEMP LAPSE RATE AT SUNRISE IN A VALLEY ATMOSPHERE.
C  THIS INFORMATION IS USED BY SUBROUTINE DESCNT IN CALCULATING THE
C  INVERSION DESTRUCTION IN A VALLEY ATMOS.  THE VALUES ARE COMPUTED
C  USING THE NEAREST UPPER AIR STATION DATA.  PRESSURE AND DENSITY ARE
C  COMPUTED BY USING THE LAYER-AVERAGED VALUES FROM THE SOUNDING WHERE
C  THE LAYER CORRESPONDS WITH THE MSL HEIGHT OF THE VALLEY FLOOR AND
C  RIDGETOPS.  IF THE VALLEY IS LOWER THAN THE SOUNDING'S LOWEST
C  OBSERVATION, THEN THE PRESSURE AND DENSITY ARE COMPUTED FROM THE
C  LOWEST OBSERVATION.  THE POTENTIAL TEMP LAPSE RATE IN THE VALLEY
C  IS COMPUTED FROM THE LAPSE RATE OF THE SND THROUGH THE LAYER FROM
C  GROUND-LEVEL UP TO 200 METERS AGL.  THEN THE (GAMMA VALLEY) = (GAMMA
C  SOUNDING)/(GAMMA SND MAX) * (GAMMA VALLEY MAX).   THE (GAMMA SND MAX)
C  IS ASSUMED TO BE 0.04 [DEG K/M] FROM C.D. WHITEMAN THESIS (ATMOS. SCI.
C  PAPER NO. 328, COLORADO STATE UNIVERSITY, FORT COLLINS, CO).
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C       NOBS = NUMBER OF OBSERVATIONS IN SOUNDING
C       HTO = ARRAY OF OBSERVATION HEIGHTS (M MSL)
C       PRO = ARRAY OF PRESSURE OBSERVATIONS (MB)
C       TMO = ARRAY OF TEMPERATURE OBSERVATIONS (DEG C)
C       PTMO = ARRAY OF POT. TEMP. OBSERVATIONS (DEG K)
C       TOSEG = HEIGHT OF VALLEY TOP (M MSL)
C       ELSEG = HEIGHT OF VALLEY FLOOR (M MSL)
C       VGAMAX = MAXIMUM LAPSE RATE IN THE VALLEY (DEGK/M)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C       PRES = AVERAGE PRESSURE THRU VALLEY DEPTH (MB)
C       RHO = AVERAGE DENSITY THRU VALLEY DEPTH (KG/M**3)
C       GAMA = POT. TEMP. LAPSE RATE THRU VALLEY DEPTH (DEG K/M)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C       COEFF
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C       LAY, TPROF
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C       (NONE)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETER DEFINITIONS:
C       NGA = NUMBER OF GAMMA (NOT P.T. LAPSE RATE) SURFACES
C       NOB = NUMBER OF OBSERVATIONS IN SOUNDING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE VALGAM(NOBS,HTO,PRO,TMO,PTMO,TOSEG,ELSEG,PRES,
     +                  RHO,VGAMAX,GAMA)
C
      PARAMETER
     +    NOB=30,     NGA=9
C
      DIMENSION
     +    HTO(NOB),   PRO(NOB),   TMO(NOB),   PTMO(NOB),
     +    TOPL(NGA),  AVE(NGA)
C
C               *COMPUTE AVERAGE TEMP AND PRESS IN VALLEY FROM
C               *NEAREST STATION MORNING SOUNDING.
      IF(TOSEG.LE.HTO(1)) THEN
        TAVE=TMO(1)+273.15
```

```
063             PAVE=PRO(1)
064           ELSE
065             IF(ELSEG.LE.HTO(1)) THEN
066               NLAYS=1
067               TOPL(1)=TOSEG
068               CALL LAY(NOBS,HTO,TMO,AVE,NLAYS,TOPL)
069               TAVE=AVE(1)+273.15
070               CALL LAY(NOBS,HTO,PRO,AVE,NLAYS,TOPL)
071               PAVE=AVE(1)
072             ELSE
073               NLAYS=2
074               TOPL(2)=TOSEG
075               TOPL(1)=ELSEG
076               CALL LAY(NOBS,HTO,TMO,AVE,NLAYS,TOPL)
077               TAVE=AVE(2)+273.15
078               CALL LAY(NOBS,HTO,PRO,AVE,NLAYS,TOPL)
079               PAVE=AVE(2)
080             ENDIF
081           ENDIF
082     C
083     C               *COMPUTE DENSITY
084           PRES=PAVE
085           RHO=PAVE/(TAVE*2.8704)
086     C
087     C               *DETERMINE GAMA FOR VALLEY ATMOSPHERE.
088           DH=200.
089           DHMSL=HTO(1)+DH
090           CALL TPROF(NOBS,HTO,PTMO,DHMSL,DPT)
091           GSTA=(DPT-PTMO(1))/DH
092           GAMA=VGAMAX*(GSTA/0.04)
093           IF(GAMA.GT.VGAMAX) GAMA=VGAMAX
094           IF(GAMA.LT.0.0001) GAMA=0.0001
095     C
096           RETURN
097           END
```

```
C  SUBROUTINE: MET.VREAL
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO COMPUTE THE UNTRANSFORMED VELOCITY
C  COMPONENTS FOR ANY ONE POINT ON A GIVEN 'GAMMA' SURFACE.
C  EACH OF THE THREE VELOCITY COMPONENTS ARE FOR THE 'X', 'Y'
C  AND 'GAMMA' COORDINATES.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C    THRU ARGUMENT:
C      FI1 = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR A GAMMA SURFACE
C      GI1 = ARRAY OF AMPLITUDE FUNCTION 'GSUBI' FOR A GAMMA SURFACE
C      GMF1 = ARRAY OF AMPL FNCT 'GSUBI-FSUBI' FOR A GAMMA SURFACE
C      BI1 = ARRAY OF POLY BASIS FUNCTIONS 'BSUBI' FOR A GAMMA SRFC
C      CI1 = ARRAY OF POLY BASIS FUNCTIONS 'CSUBI' FOR A GAMMA SRFC
C      PBI1 = ARRAY OF POLY BASIS FNCTS 'PARTIALB' FOR A GAMMA SRFC
C      RRO = AIR DENSITY AT POINT (KG/M**3)
C      DZA = TERRAIN SLOPE IN ALPHA-DIRECTION (M)
C      DZB = TERRAIN SLOPE IN BETA-DIRECTION (M)
C      NTERMS = NUMBER OF POLYNOMIAL TERMS DESCRIBING WINDS
C      PSF = GAMMA-COORDINATE OF POINT
C      DHA = SLOPE IN ALPHA-DIRECTION OF TOP BOUNDARY (M)
C      DHB = SLOPE IN BETA-DIRECTION OF TOP BOUNDARY (M)
C      HZS = 'SMALL H' (M)
C      SLX = DOMAIN X-DIMENSION (KM)
C      RXY = RATIO OF DOMAIN X-DIMENSION TO Y-DIMENSION
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C    THRU ARGUMENT:
C      RU1 = X-COMPONENT OF VELOCITY (M/S)
C      RV1 = Y-COMPONENT OF VELOCITY (M/S)
C      RW1 = GAMMA-COMPONENT OF VELOCITY (/S)
C      RWT = GAMMA-COMP OF VELOCITY SUFFICIENT TO FOLLOW TERRAIN (/S)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C      FROUDE, WINDS
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C      (NONE)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C      NONE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C      NPO = NUMBER OF POLYNOMIAL TERMS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE VREAL (FI1,GI1,GMF1,BI1,CI1,PBI1,RRO,DZA,DZB,NTERMS,
     +                  PSF,DHA,DHB,HZS,SLX,RXY,RU1,RV1,RW1,RWT)
C
      PARAMETER
     +    NPO=21
C
      DIMENSION
     +    BI1(NPO),      CI1(NPO),      FI1(NPO),      GI1(NPO),
     +    GMF1(NPO),     PBI1(NPO)
C
C
      SL=SLX*1000.
C
C             **COMPUTE BIG U, BIG V AND BIG W.
      U=0.0
```

```
      V=0.0
      W=0.0
C
      DO 1 J=1,NTERMS
        U=FI1(J)*BI1(J)+U
        V=GI1(J)*CI1(J)+V
        W=GMF1(J)*PBI1(J)+W
    1 CONTINUE
C
C               **COMPUTE REAL VELOCITY COMPONENTS.
      RU1=U/(RRO*HZS)
      RV1=V/(RXY*RRO*HZS)
      ZDS=DZA*RU1+RXY*DZB*RV1
      HD=DHA*RU1+RXY*DHB*RV1
      RW1=(W+RRO*ZDS+RRO*PSF*HD)/(RRO*SL)
C               ***COMPUTE VERTICAL VELOCITY SUFFICIENT TO FOLLOW TERRAIN.
      RWT=(ZDS+PSF*HD)/SL
C
      RETURN
      END
```

```
C  SUBROUTINE: MET.WADJ
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO CORRECT THE ORIGINAL INPUT WINDS
C  FROM BEING REFERENCED TO THE ACTUAL TERRAIN SURFACE TO BEING
C  REFERENCED TO THE FROUDE LOWER TERRAIN SURFACE (ARRAY THG).
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C    THRU ARGUMENT:
C      NFS = NUMBER OF 'STATIONS' FOR COMPUTING WIND FIELD
C      NGAMA = NUMBER OF 'GAMMA' SURFACES TO COMPUTE WINDS ON
C      HTGAMA = ARRAY OF HEIGHTS OF 'GAMMA' SURFACES (TRANSF. COORD)
C      HA = HEIGHT OF THE ADIABATIC ATMOSPHERE (M MSL)
C      ROA = SEA-LEVEL AIR DENSITY (KG/M**3)
C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
C            1 - 'SMALL H' EQUAL TO CONSTANT
C            2 - 'BIG H' EQUAL TO CONSTANT
C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
C            (SMALL H = BIG H - TERRAIN HEIGHT)
C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
C           TO HH)
C    THRU COMMON BLOCK 'AMPL1':
C      UP = ARRAY OF X-COMP OF WINDS AT EACH 'GAMMA' SURFACE FOR ALL
C           INITIAL INPUT WIND STATIONS (M/S)
C      VP = SAME AS 'UP' EXCEPT Y-COMP OF WINDS
C      XG = ARRAY OF 'ALPHA' COORD OF INPUT WIND STATIONS
C      YG = ARRAY OF 'BETA' COORD OF INPUT WIND STATIONS
C      ZSTER = ARRAY OF INPUT STATION ELEVATIONS (M MSL)
C    THRU COMMON BLOCK 'FROUD':
C      NXCF = NUMBER OF GRID CELLS IN X-DIR OF FROUDE GRID
C      NYCF = NUMBER OF GRID CELLS IN Y-DIR OF FROUDE GRID
C      THG = ARRAY OF FROUDE GRID POINT ELEVS (M MSL)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C    THRU COMMON BLOCK 'AMPL1':
C      SAME AS DESCRIBED IN 'INPUT' EXCEPT ORIGINAL INPUT STATIONS
C      (FIRST NFS POSITIONS) HAVE BEEN CORRECTED
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C      FROUDE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C      SMALLH
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C      AMPL1, FROUD
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C      NAL = NUA+NNS+NOT+(NPF*NPF)
C      NGA = NUMBER OF 'GAMMA' SURFACES
C      NPF = NUMBER OF GRID POINTS IN FROUDE NUMBER COMPUTATION
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WADJ (NFS,NGAMA,HTGAMA,HA,ROA,HH,IHH)
C
      PARAMETER
     +    NGA=9,          NAL=155,        NPF=10
C
      DIMENSION
```

```
     +     HTGAMA(NGA)
C
      COMMON /AMPL1/
     +     UP(NAL,NGA),    VP(NAL,NGA),    XG(NAL),        YG(NAL),
     +     ZSTER(NAL)
C
      COMMON /FROUD/
     +     AG(NPF),        BG(NPF),        OH(NPF,NPF),    THET(NPF,NPF),
     +     THG(NPF,NPF),   XGF(NPF),       YGF(NPF),       NXCF,
     +     NYCF
C
      XCF=REAL(NXCF)
      YCF=REAL(NYCF)
C
      DO 20 NS=1,NFS
C           **CALCULATE ARRAY INDICIES IN FROUDE TERRAIN FILE.
        IF(XG(NS).LE.0.) THEN
          II=1
        ELSE
          IF(XG(NS).GE.1.) THEN
            II=NXCF
          ELSE
            II=NINT(XCF*XG(NS)-0.5)+1
          ENDIF
        ENDIF
        IF(YG(NS).LE.0.) THEN
          JJ=1
        ELSE
          IF(YG(NS).GE.1.) THEN
            JJ=NYCF
          ELSE
            JJ=NINT(YCF*YG(NS)-0.5)+1
          ENDIF
        ENDIF
C           ***COMPUTE 'SMALLH' ON ACTUAL TERRAIN AND FROUD TERRAIN
        ZAC=ZSTER(NS)
        CALL SMALLH(IHH,HH,ZAC,XG(NS),YG(NS),HAC)
        ZF=THG(II,JJ)
        CALL SMALLH(IHH,HH,ZF,XG(NS),YG(NS),HF)
        ZSTER(NS)=ZF
C           ***CORRECT WINDS ON GAMMA SURFACES.
        DO 10 NG=1,NGAMA
          Z=ZAC+HTGAMA(NG)*HAC
          RO=ROA*(1.-Z/HA)**2.5
          ROZAC=RO*HAC
          Z=ZF+HTGAMA(NG)*HF
          RO=ROA*(1.-Z/HA)**2.5
          ROZF=RO*HF
          RAT=ROZAC/ROZF
          UP(NS,NG)=RAT*UP(NS,NG)
          VP(NS,NG)=RAT*VP(NS,NG)
   10   CONTINUE
   20 CONTINUE
C
      RETURN
      END
```

```
C  SUBROUTINE: MET.WARM
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE CALCULATES THE ENERGY (JOULES/M**2) GONE INTO
C  WARMING THE AIR BETWEEN THE MORNING AND AFTERNOON SOUNDING AT ONE
C  STATION.  THIS IS A MEASURE OF THE SENSIBLE HEAT FLUX WHICH IS
C  USED IN COMPUTING THE VALLEY INVERSION DESTRUCTION.  THE ENERGY
C  IS COMPUTED BY INTEGRATING UP THE SOUNDINGS THE QUANTITY
C  PRES/PT*DELTA(PT) AND THEN MULTIPLYING BY THE QUANTITY CP/R
C  WHERE, PRESS IS THE AVERAGE PRESSURE BETWEEN THE SOUNDINGS, PT
C  IS THE AVERAGE POTENTIAL TEMP BETWEEN THE SOUNDINGS,
C  DELTA(PT) IS THE DIFFERENCE IN POTENTIAL TEMP BETWEEN THE TWO
C  SOUNDINGS, CP IS THE SPECIFIC HEAT AT CONSTANT PRESSURE, AND
C  R IS THE UNIVERSAL GAS CONSTANT.  WARM OR COLD AIR ADVECTION IS
C  REMOVED BY SHIFTING THE AFTERNOON POTENTIAL TEMP SOUNDING TO
C  MATCH THE MORNING SOUNDING IN THE UPPER PORTION OF THE MORNING
C  SOUNDING.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C       NOBM = NUMBER OF OBSERVATIONS IN MORNING SOUNDING
C       NOBA = NUMBER OF OBSERVATIONS IN AFTERNOON SOUNDING
C       THTM = ARRAY OF OBS. HEIGHTS FOR MORN. SND. (M MSL)
C       THTA = ARRAY OF OBS. HEIGHTS FOR AFT. SND (M MSL)
C       PHTM = ARRAY OF PRES. OBS. FOR MORN. SND. (MB)
C       PHTA = ARRAY OF PRES. OBS. FOR AFT. SND. (MB)
C       PTM = ARRAY OF POT. TEMP. OBS. FOR MORN. SND. (DEG K)
C       PTA = ARRAY OF POT. TEMP. OBS. FOR AFT. SND. (DEG K)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C       ENER = MEASURE OF SENSIBLE HEAT FLUX (JOULES/M**2)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CONSTANT DEFINITIONS:
C       CONST = 350.149  [100.*CP/R WHERE CP IS THE SPECIFIC HEAT AT
C                         CONSTANT PRESSURE (1005 J/KG/DEGK), AND
C                         R IS THE GAS CONSTANT (287.021 J/KG/DEGK);
C                         THE FACTOR 100 IS IN THE UNITS CONVERSION]
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C       COEFF
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C       TPROF
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C       (NONE)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETER DEFINITIONS:
C       NOB = TWICE MAX. NUMBER OF OBS. EXPECTED IN SOUNDING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WARM(NOBM,NOBA,THTM,THTA,PHTM,PHTA,PTM,PTA,ENER)
C
      PARAMETER NOB=30
C
      LOGICAL POS
C
      DIMENSION THTM(NOB),THTA(NOB),PHTM(NOB),PHTA(NOB),PTM(NOB),
     +PTA(NOB),THTMP(NOB),THTAP(NOB),PHTMP(NOB),PHTAP(NOB),
     +PTMP(NOB),PTAP(NOB)
C
C                ***CONSTANTS
      CONST=350.149
```

```
C
C                ***SPECIFY MAXIMUM HEIGHT TO INTEGRATE THROUGH AS
C                ***3000 METERS ABOVE GROUND-LEVEL.
      UPBND=THTM(1)+3000.
C
      NOBMP=NOBM
      NOBAP=NOBA
      DO 1 J=1,NOBM
      THTMP(J)=THTM(J)
      PHTMP(J)=PHTM(J)
      PTMP(J)=PTM(J)
    1 CONTINUE
      DO 2 J=1,NOBA
      THTAP(J)=THTA(J)
      PHTAP(J)=PHTA(J)
      PTAP(J)=PTA(J)
    2 CONTINUE
C
C                ***SPECIFY VALUES IN MORNING SOUNDING AT AFTERNOON
C                ***SOUNDING OBSERVATION HEIGHTS.
      DO 40 N=2,NOBAP
        DO 35 M=2,NOBMP
          IF(THTAP(N).EQ.THTMP(M)) GO TO 40
          IF(THTAP(N).LT.THTMP(M)) THEN
            CALL TPROF(NOBMP,THTMP,PTMP,THTAP(N),PTVAL)
            CALL TPROF(NOBMP,THTMP,PHTMP,THTAP(N),PHTVAL)
            NOBMP=NOBMP+1
            DO 30 J=NOBMP,M+1,-1
              THTMP(J)=THTMP(J-1)
              PHTMP(J)=PHTMP(J-1)
              PTMP(J)=PTMP(J-1)
   30       CONTINUE
            THTMP(M)=THTAP(N)
            PHTMP(M)=PHTVAL
            PTMP(M)=PTVAL
            GO TO 40
          ENDIF
   35   CONTINUE
   40 CONTINUE
C
C                ***SPECIFY VALUES IN AFTERNOON SOUNDING AT MORNING
C                ***SOUNDING OBSERVATION HEIGHTS.
      DO 60 N=2,NOBMP
        DO 55 M=2,NOBAP
          IF(THTMP(N).EQ.THTAP(M)) GO TO 60
          IF(THTMP(N).LT.THTAP(M)) THEN
            CALL TPROF(NOBAP,THTAP,PTAP,THTMP(N),PTVAL)
            CALL TPROF(NOBAP,THTAP,PHTAP,THTMP(N),PHTVAL)
            NOBAP=NOBAP+1
            DO 50 J=NOBAP,M+1,-1
              THTAP(J)=THTAP(J-1)
              PHTAP(J)=PHTAP(J-1)
              PTAP(J)=PTAP(J-1)
   50       CONTINUE
            THTAP(M)=THTMP(N)
            PHTAP(M)=PHTVAL
            PTAP(M)=PTVAL
            GO TO 60
          ENDIF
   55   CONTINUE
   60 CONTINUE
C
```

```
125     C               ***DETERMINE TOP OBSERVATION POINT AND VALUES AT
126     C               ***TOP OBSERVATION POINT.
127           HTMAX=AMIN1(THTMP(NOBMP),THTAP(NOBAP),UPBND)
128     C               *MORNING SOUNDING
129           DO 5 J=1,NOBMP
130             IF(THTMP(J).EQ.HTMAX) THEN
131               NOBS=J
132               GO TO 6
133             ENDIF
134             IF(THTMP(J).GT.HTMAX) THEN
135               CALL TPROF(NOBMP,THTMP,PTMP,HTMAX,PTMP(J))
136               CALL TPROF(NOBMP,THTMP,PHTMP,HTMAX,PHTMP(J))
137               THTMP(J)=HTMAX
138               NOBS=J
139               GO TO 6
140             ENDIF
141         5 CONTINUE
142         6 CONTINUE
143     C               *AFTERNOON SOUNDING
144           DO 7 J=1,NOBAP
145             IF(THTAP(J).EQ.HTMAX) THEN
146               NOBS=J
147               GO TO 8
148             ENDIF
149             IF(THTAP(J).GT.HTMAX) THEN
150               CALL TPROF(NOBAP,THTAP,PTAP,HTMAX,PTAP(J))
151               CALL TPROF(NOBAP,THTAP,PHTAP,HTMAX,PHTAP(J))
152               THTAP(J)=HTMAX
153               NOBS=J
154               GO TO 8
155             ENDIF
156         7 CONTINUE
157         8 CONTINUE
158     C
159     C               ***DETERMINE LARGEST NEGATIVE POTENTIAL TEMPERATURE
160     C               ***DIFFERENCE AND SMALLEST POSITIVE POTENTIAL TEMP.
161     C               ***DIFFERENCE BETWEEN MORNING AND AFTERNOON SOUNDINGS.
162           POS=.TRUE.
163           PTNEG=0.
164           PTPOS=100.
165           DO 65 N=1,NOBS
166             DIF=PTAP(N)-PTMP(N)
167             IF(DIF.LE.0.) THEN
168               POS=.FALSE.
169               IF(DIF.LE.PTNEG) PTNEG=DIF
170             ELSE
171               IF(POS.AND.DIF.LE.PTPOS) PTPOS=DIF
172             ENDIF
173        65 CONTINUE
174     C
175     C               ***ADJUST AFTERNOON SOUNDING FOR ADVECTION.
176           ADJ=PTNEG
177           IF(POS) ADJ=PTPOS
178           DO 20 J=1,NOBS
179             PTAP(J)=PTAP(J)-ADJ
180        20 CONTINUE
181     C
182     C               ***INTEGRATE UP SOUNDING
183           X=0.
184           DO 70 J=2,NOBS
185           DELTAZ=THTMP(J)-THTMP(J-1)
186           PT1=(PTMP(J)+PTMP(J-1))/2.
```

```
.87           PT2=(PTAP(J)+PTAP(J-1))/2.
.88           PTAVE=(PT1+PT2)/2.
.89           DELTPT=PT2-PT1
.90           PHT1=(PHTMP(J)+PHTMP(J-1))/2.
.91           PHT2=(PHTAP(J)+PHTAP(J-1))/2.
.92           PHTAVE=(PHT1+PHT2)/2.
.93           X=DELTAZ*DELTPT*PHTAVE/PTAVE+X
.94        70 CONTINUE
.95           ENER=CONST*X
.96     C
.97     C     RETURN
.98           END
```

```
C  SUBROUTINE: MET.WINDS
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO COMPUTE THE WINDS AT A POINT  GIVEN
C  THE LOCATION OF THE POINT, THE AMPLITUDE FUNCTIONS AND THE
C  TERRAIN DATA.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C    THRU ARGUMENT:
C      A - ALPHA COORDINATE OF POINT TO COMPUTE WINDS.
C      B - BETA COORDINATE OF POINT.
C      Z - VERTICAL COORDINATE OF POINT (M AGL).
C      NTERMS - NUMBER OF POLYNOMIAL TERMS.
C      NGAMA - NUMBER OF GAMMA SURFACES.
C      HTGAMA - ARRAY OF HEIGHTS OF GAMMA SURFACES (0. TO 1.)
C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
C            1 - 'SMALL H' EQUAL TO CONSTANT
C            2 - 'BIG H' EQUAL TO CONSTANT
C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
C            (SMALL H = BIG H - TERRAIN HEIGHT)
C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
C           TO HH)
C      SLX = X-DIRECTION DISTANCE ACROSS DOMAIN (KM)
C      RXY = RATIO OF X-DIMENSION OF DOMAIN TO Y-DIMENSION OF DOMAIN
C    THRU COMMON BLOCK 'AMPLFN':
C      FI = -----I
C      GI =      I --- AMPL FNCTS FOR ALL 'GAMMA' SURFACES
C      GMF = ----I
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C    THRU ARGUMENT:
C      U - COMPONENT OF WINDS IN X-DIRECTION (M/SEC).
C      V - COMPONENT OF WINDS IN Y-DIRECTION (M/SEC).
C      W - COMPONENT OF WINDS IN Z-DIRECTION (M/SEC).
C      WT - COMPONENT OF WINDS IN Z-DIRECTION SUFFICIENT ONLY
C           TO FOLLOW TERRAIN (M/SEC).
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C      BLPARS, MIXSTB
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C      POLBAS, SHSLP, SMALLH, TETRI, VREAL
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C      AMPLFN
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C      NPO = NUMBER OF POLYNOMIAL TERMS
C      NGA = NUMBER OF 'GAMMA' SURFACES
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WINDS(A,B,Z,NTERMS,NGAMA,HTGAMA,HA,ROA,HH,IHH,SLX,RXY,
     +                 U,V,W,WT)
C
      PARAMETER
     +    NGA=9,          NPO=21
C
      DIMENSION
     +    BI(NPO),        CI(NPO),        FI1(NPO),       GI1(NPO),
```

```
     +    GMF1(NPO),     HTGAMA(NGA),   LA(2),          PBI(NPO),
     +    RU(2),         RV(2),         RW(2),         RWT(2)
C
      COMMON /AMPLFN/
     +    FI(NPO,NGA),   GI(NPO,NGA),   GMF(NPO,NGA)
C
      CALL TETRI (A,B,TH,DZA,DZB)
      CALL POLBAS (A,B,NTERMS,BI,CI,PBI)
      CALL SHSLP (IHH,DZA,DZB,A,B,DHA,DHB)
      CALL SMALLH (IHH,HH,TH,A,B,HZS)
C
C            **DEFINE TWO "GAMMA" SURFACES TO INTERPOLATE WINDS
C            **BETWEEN.
      G=Z/HZS
      IF (G.GT.1.0)  THEN
        WRITE (6,10)
        STOP 'WINDS'
      ENDIF
      IF (G.LT.HTGAMA(1)) G=HTGAMA(1)
      IF (G.EQ.HTGAMA(NGAMA)) GO TO 3
      IF (G.EQ.HTGAMA(1)) GO TO 4
C
      DO 1 K=NGAMA-1,1,-1
        IF (G.GE.HTGAMA(K)) GO TO 2
    1 CONTINUE
      WRITE (6,11)
      STOP 'WINDS'
C
    2 LA(1)=K
      LA(2)=K+1
      GO TO 5
C
    3 LA(1)=NGAMA
      LA(2)=NGAMA
      GO TO 5
C
    4 LA(1)=1
      LA(2)=1
      GO TO 5
C
    5 CONTINUE
C
C            **COMPUTE WIND COMPONENTS FOR TWO SURFACES .
      DO 7 NL=1,2
        ZL=TH-HTGAMA(LA(NL))*HZS
        RO=ROA*(1.-ZL/HA)**2.5
C
        DO 6 NN=1,NTERMS
          FI1(NN)=FI(NN,LA(NL))
          GI1(NN)=GI(NN,LA(NL))
          GMF1(NN)=GMF(NN,LA(NL))
    6   CONTINUE
        CALL VREAL (FI1,GI1,GMF1,BI,CI,PBI,RO,DZA,DZB,NTERMS,
     +  HTGAMA(LA(NL)),DHA,DHB,HZS,SLX,RXY,RU(NL),RV(NL),RW(NL),RWT(NL))
    7 CONTINUE
C
C            **SPATIALLY INTERPOLATE WINDS.
      IF (LA(2).EQ.LA(1)) GO TO 8
      S2=(G-HTGAMA(LA(1)))/(HTGAMA(LA(2))-HTGAMA(LA(1)))
      S1=1.-S2
      GO TO 9
C
```

```
125         8 S2=1.
126           S1=0.
127       C
128         9 CONTINUE
129       C
130           U=S2*RU(2)+S1*RU(1)
131           V=S2*RV(2)+S1*RV(1)
132           W=S2*RW(2)+S1*RW(1)
133           WT=S2*RWT(2)+S1*RWT(1)
134       C
135           RETURN
136        10 FORMAT (5X,' THE GAMMA COORDINATE IS GREATER THAN 1.0 IN
137         +  SUBROUTINE WINDS.')
138        11 FORMAT (5X,' GAMMA OF POINT NOT WINTIN GAMMA DOMAIN IN
139         +  SUBROUTINE WINDS.')
140           END
```

```
C  SUBROUTINE: MET.WINGAM
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO COMPUTE THE WINDS ON ALL 'GAMMA'
C  SURFACES ABOVE A POINT ON THE TERRAIN GIVEN THE LOCATION OF THE
C  POINT, THE AMPLITUDE FUNCTIONS AND THE TERRAIN DATA.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C    THRU ARGUMENT:
C      A - ALPHA COORDINATE OF POINT TO COMPUTE WINDS.
C      B - BETA COORDINATE OF POINT.
C      TH - TERRAIN HEIGHT (M MSL).
C      DZA - SLOPE OF TERRAIN IN ALPHA DIRECTION (M)
C      DZB - SLOPE OF TERRAIN IN BETA DIRECTION (M)
C      NTERMS - NUMBER OF POLYNOMIAL TERMS.
C      NGAMA - NUMBER OF GAMMA SURFACES.
C      HTGAMA - ARRAY OF HEIGHTS OF GAMMA SURFACES (0. TO 1.)
C      HA = HEIGHT OF THE ADIABATIC ATMOSPHERE (M).
C      ROA = AIR DENSITY AT SEA-LEVEL (KG/M**3).
C      IHH = DEFINES FORM OF TOP SHEET (1, 2, 3, OR 4)
C            1 - 'SMALL H' EQUAL TO CONSTANT
C            2 - 'BIG H' EQUAL TO CONSTANT
C            3 - 'SMALL H' EQUAL TO FUNCTION OF ALPHA AND BETA
C            4 - 'BIG H' EQUAL TO FUNCTION OF ALPHA AND BETA
C            (SMALL H = BIG H - TERRAIN HEIGHT)
C      HH = VALUE FOR 'SMALL H' OF 'BIG H' DEPENDING ON WHETHER
C           IHH=1 OF IHH=2, RESPECTIVELY. (IF STATION OUTSIDE DOMAIN
C           FOR IHH=3 OF IHH=4, THEN 'SMALL H' OR 'BIG H' IS SET EQUAL
C           TO HH)
C      SLX = DOMAIN X-DIMENSION (KM)
C      RXY = RATIO OF DOMAIN X-DIMENSION TO Y-DIMENSION
C    THRU COMMON BLOCK 'AMPLFN':
C      FI = -----I
C      GI =      I --- AMPL FNCTS FOR ALL 'GAMMA' SURFACES
C      GMF = ----I
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C    THRU ARGUMENT:
C      RU = ARRAY OF X-COMPONENT OF WINDS ON ALL 'GAMMA' SRFCS (M/SEC).
C      RV = ARRAY OF Y-COMPONENT OF WINDS ON ALL 'GAMMA' SRFCS (M/SEC).
C      RW = ARRAY OF Z-COMPONENT OF WINDS ON ALL 'GAMMA' SRFCS (M/SEC).
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C      AMPLIT, FROUDE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C      POLBAS, SHSLP, SMALLH, VREAL
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C      AMPLFN
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  PARAMETERS DEFINITIONS:
C      NPO = NUMBER OF POLYNOMIAL TERMS
C      NGA = NUMBER OF 'GAMMA' SURFACES
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WINGAM (A,B,TH,DZA,DZB,NTERMS,NGAMA,HTGAMA,HA,ROA,
     +                   HH,IHH,SLX,RXY,RU,RV,RW)
C
      PARAMETER
     +    NGA=9,          NPO=21
C
```

```
      DIMENSION
     +    BI(NPO),        CI(NPO),        FI1(NPO),       GI1(NPO),
     +    GMF1(NPO),      HTGAMA(NGA),    PBI(NPO),       RU(NGA),
     +    RV(NGA),        RW(NGA)
C
      COMMON /AMPLFN/
     +    FI(NPO,NGA),    GI(NPO,NGA),    GMF(NPO,NGA)
C
      CALL POLBAS (A,B,NTERMS,BI,CI,PBI)
      CALL SHSLP (IHH,DZA,DZB,A,B,DHA,DHB)
      CALL SMALLH (IHH,HH,TH,A,B,HZS)
C
C             **COMPUTE WIND COMPONENTS ON EACH GAMMA SURFACE.
      DO 7 NL=1,NGAMA
        ZL=TH+HTGAMA(NL)*HZS
        RO=ROA*(1.-ZL/HA)**2.5
C
        DO 6 NN=1,NTERMS
          FI1(NN)=FI(NN,NL)
          GI1(NN)=GI(NN,NL)
          GMF1(NN)=GMF(NN,NL)
    6   CONTINUE
        CALL VREAL (FI1,GI1,GMF1,BI,CI,PBI,RO,DZA,DZB,NTERMS,
     +  HTGAMA(NL),DHA,DHB,HZS,SLX,RXY,RU(NL),RV(NL),RW(NL),RWT)
    7 CONTINUE
C
      RETURN
      END
```

```
C  SUBROUTINE: MET.WNDRSP
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO COMPUTE THE WIND DIRECTION AND SPEED
C  GIVEN THE VELOCITY COMPONENTS.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C       U = X-COMPONENT OF VELOCITY (M/S)
C       V = Y-COMPONENT OF VELOCITY (M/S)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C       WS = WIND SPEED (M/S)
C       WD = WIND DIRECTION (DEG)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C       AMPLIT, FROUDE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C       (NONE)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C       (NONE)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CONSTANTS DEFINITIONS:
C       RTOD = NUMBER OF DEGREES PER RADIAN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WNDRSP (U,V,WD,WS)
C
      RTOD=57.29583
C
      WS=SQRT(U**2+V**2)
      IF (WS.EQ.0.) THEN
        WD=0.
      ELSE
        IF(U.GE.0..AND.V.GE.0.) WD=270.-ASIN(V/WS)*RTOD
        IF(U.LT.0..AND.V.LE.0.) WD=ASIN(ABS(U)/WS)*RTOD
        IF(U.EQ.0..AND.V.LE.0.) WD=360.
        IF(U.GE.0..AND.V.LE.0.) WD=270.+ACOS(U/WS)*RTOD
        IF(U.LE.0..AND.V.GE.0.) WD=90.+ASIN(V/WS)*RTOD
      ENDIF
C
      RETURN
      END
```

```
C  SUBROUTINE: MET.WSTAR
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO COMPUTE THE CONVECTIVE VELOCITY
C  SCALE W* USING THE APPROACH GIVEN IN MESOPUFF II (MARCH 1984).
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE INPUTS ARE:
C      TSURF = AIR SURFACE TEMPERATURE (DEGK)
C      WT = W'THETA' ((M-DEGK)/S)
C      ZI = MIXING HEIGHT (M AGL)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  THE OUTPUTS ARE:
C      WSTR = CONVECTIVE VELOCITY SCALE (M/S)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CONSTANT DEFINITIONS:
C      GRAV = 9.81  [ACCELERATION DUR TO GRAVITY (M/S**2)]
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLING SUBROUTINES:
C      BLPARS
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CALLED SUBROUTINES:
C      (NONE)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  COMMON BLOCKS:
C      (NONE)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WSTAR(TSURF,WT,ZI,WSTR)
C
C             ***CONSTANTS
      GRAV=9.81
C
C             ***CALC. CONVECTIVE VELOCITY SCALE, (CALCULATED ONLY
C             ***DURING CONVECTIVE CONDITIONS, I.E. W'THETA' > 0)
      IF(WT.LE.0.) THEN
        WSTR=0.
      ELSE
        WSTR=(GRAV*WT*ZI/TSURF)**0.3333333
      ENDIF
C
      RETURN
      END
```

APPENDIX Q
FORTRAN LISTING OF POLUT

```
001     C  PROGRAM: POLUT.AAPOL        (VERSION 1.0)
002     C  DATE: APRIL 1985
003     C
004     C    THIS PROGRAM IS USED TO COMPUTE POLLUTANT CONCENTRATIONS FOR
005     C  UP TO 5 POINT SOURCES AND TWO POLLUTANTS.  UP TO 4 RECEPTOR GRIDS
006     C  AND 10 INDIVIDUAL RECEPTORS CAN BE SPECIFIED.  SOURCES LOCATED IN
007     C  VALLEYS ARE GIVEN SPECIAL TREATMENT.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C    THE PUFF CODE USED TO IDENTIFY WHICH PUFFS FORM QUADRILATERALS
010     C    AND WHICH PUFFS FORM SEGMENTS IS:
011     C      IQCOD = 0 - A STARTING PUFF.
012     C            = 1 - A MIDDLE PUFF IN THE CHAIN.
013     C            = 2 - AN ENDING PUFF.
014     C            = 3 - A LONER PUFF (NOT USED).
015     C            = 4 - PUFF OFF GRID.
016     C            = 5 - PUFF AT THE SOURCE.
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C    CONSTRAINTS:
019     C       1. SIMULATION PERIOD MUST BE INCLUDED WITHIN MET DATA PERIOD.
020     C       2. SIMULATION PERIOD MUST BE AT LEAST THREE HOURS LONG.
021     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
022     C
023           PARAMETER
024          +    NGA=9,          NPO=21,         NPU=100,        NSC=5,
025          +    NPG=10,         NSG=12,         NVA=5
026     C
027           DIMENSION
028          +    IS(NPG),        CLKD(NSC),      NSGA(NSC)
029     C
030           LOGICAL
031          +    LUFORM,         WOFF,           VALL,           OFFS(NSC),
032          +    CONT(NSC),      FLAG(NSC)
033     C
034           COMMON /MAIN2/
035          +    HA,             HH,             IHH,            IXO,
036          +    IYO,            ROA,            RXY,            SLX,
037          +    SLY
038     C
039           COMMON /CLOCK1/
040          +    CLK1,           CLK2,           CLK3
041     C
042           COMMON /WINDS1/
043          +    FI(NPO,NGA,3), GI(NPO,NGA,3), GMF(NPO,NGA,3), HTGAMA(NGA),
044          +    THETTA(3),     NGAMA,          NTERMS(3)
045     C
046           COMMON /BLPAR1/
047          +  BLP(NPG,NPG,3,7), XCS,           YCS,            HCELLX,
048          +  HCELLY,           NXCS,          NYCS
049     C
050           COMMON /VALLY1/
051          +   COEF(NSG,NVA,3),COXV(NVA),      COYV(NVA),      WSV(NVA),
052          +   SEGX(NSG,NVA),  SEGY(NSG,NVA), NSEG(NVA),      IDVAL(NVA),
053          +   NVALE,          QM(NSG,NSC,2), VSEGL(NSG,NVA),WIDV(NSG,NVA)
054     C
055           COMMON /PUFF1/
056          +   A(NPU,NSG,NSC),      B(NPU,NSG,NSC),      IPCOD(NPU,NSG,NSC),
057          +   SGPRY(NPU,NSC),     SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
058          +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
059          +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
060          +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),     ZMIX(NPU,NSG,NSC),
061          +   QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
062          +   NPUFFV(NSC)
```

```
063     C
064           COMMON /TPDT/
065          +     ANQ(2),          BNQ(2),          DSMQ(2),         HTMIXQ(2),
066          +     IPCODQ(2),       SYDPQ(2),        SYDGQ(2),        SYSHQ(2),
067          +     SYTOQ(2),        SZDQ(2),         SZTOQ(2),        ZNQ(2),
068          +     ANQS(2),         BNQS(2),         DSMQS(2),        ZNQS(2),
069          +     AF(4),           BF(4),           QF(2)
070     C
071           COMMON /REC/
072          +     NINDR,           NPTSX(4),        NPTSY(4),        NRGRID,
073          +     RSPAC(4),        STAX(4),         STAY(4),         XIREC(10),
074          +     XREC(25,4),      YIREC(10),       YREC(25,4),      ZIREC(10),
075          +     ZREC(25,25,4)
076     C
077           COMMON /CONC1/
078          +     CHI(25,25,4,2),CHIS(10,2)
079     C
080           COMMON /RUNSP/
081          +     LUFORM,          METHOD,          WOFF
082     C
083           COMMON /SOURC1/
084          +     EMIS(NSC,2),     NSOURC,          RS(NSC),         TS(NSC),
085          +     VS(NSC),         XSL(NSC),        YSL(NSC),        ZSH(NSC),
086          +     ELS(NSC),        AS(NSC),         BS(NSC),         SIGIN(NSC),
087          +     IDSVAL(NSC),     VALL(NSC),       INDXV(NSC)
088     C
089     C               ***INITIALIZE RUN
090           CALL INIT(ISTART,ISTOPT,IPRR,NPLT,LUO)
091     C
092           NGGGD=100
093           STDF=1013./293.16
094           RIX0=REAL(IX0)*1000.
095           RIY0=REAL(IY0)*1000.
096           SLX1=SLX*1000.
097           SLY1=SLY*1000.
098           NHOURS=ISTOPT-ISTART
099           DELTAT=1./REAL(IPRR)
100     C
101     C               ***INITIALIZE STARTING NUMBER OF PUFFS FOR EACH SOURCE
102     C               ***AND DETERMINE WHETHER VALLY SRCS ON OR OFF AT START.
103           DO 26 NS=1,NSOURC
104             NPUFFS(NS)=0
105             NPUFFV(NS)=0
106             IF(VALL(NS)) THEN
107               OFFS(NS)=.FALSE.
108               CONT(NS)=.FALSE.
109               FLAG(NS)=.FALSE.
110             ENDIF
111        26 CONTINUE
112     C
113     C               ***LOOP ON NUMBER OF HOURS IN SIMULATION
114           DO 95 NH=1,NHOURS
115           NHC=2
116           CLKIN=REAL(ISTART+NH-1)
117           IF(NH.LE.NHOURS-1) THEN
118             NHC=1
119             CLK1=CLKIN
120             CLK2=CLKIN+1.
121             CLK3=CLKIN+2.
122           ENDIF
123     C
124     C               ***LOOP ON NUMBER OF PUFFS RELEASED PER HOUR
```

```
125           DO 90 IR=1,IPRR
126           CLK=CLKIN+REAL(IR-1)*DELTAT
127     C
128     C               ***LOOP ON SOURCES.
129           DO 38 NS=1,NSOURC
130     C
131     C               ***IF NPUFFS = ARRAY SIZE, THEN PURGE PUFF ARRAYS.
132           IF(NPUFFS(NS).GE.NPU) CALL CLEAN(NS)
133           ID=INDXV(NS)
134           IF(VALL(NS).AND.NPUFFV(NS).GE.NPU) CALL CLEANV(NS,NSEG(ID))
135     C
136           NG=1
137           IF(VALL(NS)) THEN
138             IF(COEF(NSEG(ID),ID,NHC).NE.0.) THEN
139               IF(OFFS(NS)) THEN
140                 OFFS(NS)=.FALSE.
141                 CALL VALMA(NS,CLK,CLKD(NS),NSGA(NS),NPLT)
142                 FLAG(NS)=.TRUE.
143               ENDIF
144               IF(FLAG(NS).AND.IR.EQ.1) THEN
145                 CONT(NS)=.FALSE.
146                 NGB=NSEG(ID)-NSGA(NS)+1
147                 DO 19 NGG=NGB,NSEG(ID)
148                   IF(COEF(NGG,ID,NHC).LT.1.) CONT(NS)=.TRUE.
149        19         CONTINUE
150               ENDIF
151               CALL VALY(CLK,DELTAT,NSGA(NS),NS,NHC,CONT(NS),FLAG(NS),
152         +                     IPRR,NPLT)
153               NPUFFS(NS)=NPUFFS(NS)+1
154               N=NPUFFS(NS)
155               IQCOD(N,NG,NS)=5
156               GO TO 20
157             ELSE
158               CONT(NS)=.FALSE.
159               N=NPUFFS(NS)
160               IF(.NOT.OFFS(NS)) THEN
161                 OFFS(NS)=.TRUE.
162                 CLKD(NS)=CLK
163                 CALL VALY(CLK,DELTAT,NSGA(NS),NS,NHC,CONT(NS),FLAG(NS),
164         +                       IPRR,NPLT)
165                 IF(N.EQ.0) GO TO 33
166                 NPUFFS(NS)=NPUFFS(NS)+1
167                 N=NPUFFS(NS)
168                 IQCOD(N,NG,NS)=2
169                 GO TO 20
170               ENDIF
171               CALL VALY(CLK,DELTAT,NSGA(NS),NS,NHC,CONT(NS),FLAG(NS),
172         +                     IPRR,NPLT)
173               IF(N.EQ.0) GO TO 33
174               GO TO 21
175             ENDIF
176           ELSE
177             NPUFFS(NS)=NPUFFS(NS)+1
178             N=NPUFFS(NS)
179             IQCOD(N,NG,NS)=5
180           ENDIF
181     C
182     C               ***INITIALIZE CHARACTERISTICS OF STARTING PUFF FROM EACH
183     C               ***SOURCE.
184        20 A(N,NG,NS)=AS(NS)
185           B(N,NG,NS)=BS(NS)
186           TIME(N,NG,NS)=0.
```

```
187           STOTM(N,NG,NS)=1.
188           IPCOD(N,NG,NS)=1
189           DO 28 L=1,NPLT
190             QTP(N,NG,NS,L)=EMIS(NS,L)*DELTAT*3600.
191        28 CONTINUE
192           CALL WINDS (A(N,NG,NS),B(N,NG,NS),ZSH(NS),CLK,U,V,W,WT)
193           SPEED=SQRT(U**2+V**2)
194           CALL BLPINT (A(N,NG,NS),B(N,NG,NS),CLK,HMIX,ISTB,USTR,CAPL,WSTR,
195          +             TSURF,PRESS)
196           RLAPSE=.0137
197           CALL PRISE (TSURF,VS(NS),RS(NS),TS(NS),SPEED,ISTB,RLAPSE,HMIX,
198          +            ZSH(NS),RISE)
199           Z(N,NG,NS)=ZSH(NS)+RISE
200           ZSHP=Z(N,NG,NS)
201           ZMIX(N,NG,NS)=HMIX
202           IF (Z(N,NG,NS).GE.ZMIX(N,NG,NS)) IPCOD(N,NG,NS)=3
203           IF (LUFORM.AND.IPCOD(N,NG,NS).EQ.1) IPCOD(N,NG,NS)=2
204           CALL YINIT (SIGIN(NS),RISE,SGPRY(N,NS),SGDPY(N,NG,NS),
205          +            SGDGY(N,NG,NS),SGSHY(N,NG,NS),SGTOY(N,NG,NS))
206           CALL ZINIT (RISE,SGPRZ(N,NS),SGDZ(N,NG,NS),SGTOZ(N,NG,NS))
207     C
208     C                ***COMPUTE NEW LOCATION AND SIGMAS OF OLDEST PUFF.
209        21 N=1
210        29 IF (IQCOD(N,NG,NS).EQ.4)  THEN
211             N=N+1
212             IF (N.GT.NPUFFS(NS)) GO TO 33
213             GO TO 29
214           ENDIF
215           CALL NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD(N,NG,NS),ZSHP,ELS(NS),
216          +             A(N,NG,NS),B(N,NG,NS),Z(N,NG,NS),CLK,ANQ(1),BNQ(1),
217          +             ZNQ(1),IQCOD(N,NG,NS),DSMQ(1))
218           IF (IQCOD(N,NG,NS).EQ.4)  THEN
219             N=N+1
220             IF (N.GT.NPUFFS(NS)) GO TO 33
221             IF (IQCOD(N,NG,NS).EQ.1) IQCOD(N,NG,NS)=0
222             IF (IQCOD(N,NG,NS).EQ.2) IQCOD(N,NG,NS)=4
223             GO TO 29
224           ENDIF
225     C
226     C                ***COMPUTE SIGMAY AND SIGMAZ.
227           CALL SIGGS(N,NG,NS,CLK,DELTAT,1)
228     C
229           IF (IQCOD(N,NG,NS).EQ.5)  THEN
230     C               ***FIRST PUFF PROCESSING
231             ICONFG=1
232             CALL CALC(N,NG,NS,ICONFG,NPLT,DELTAT,NGGGD)
233             IQCOD(N,NG,NS)=0
234             GO TO 33
235           ENDIF
236     C
237     C                ***CALCULATE CONCENTRATION FOR REST OF PUFFS.
238        31 N=N+1
239           CALL NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD(N,NG,NS),ZSHP,ELS(NS),
240          +             A(N,NG,NS),B(N,NG,NS),Z(N,NG,NS),CLK,ANQ(2),BNQ(2),
241          +             ZNQ(2),IQCOD(N,NG,NS),DSMQ(2))
242     C
243           IF (IQCOD(N,NG,NS).EQ.4)  THEN
244             N=N-1
245             IF (IQCOD(N,NG,NS).EQ.0) IQCOD(N,NG,NS)=4
246             IF (IQCOD(N,NG,NS).EQ.1) IQCOD(N,NG,NS)=2
247             ICONFG=1
248             CALL UPDATE(N,NG,NS,ICONFG,DELTAT,NGGGD)
```

```
249             N=N+2
250             IF(N.GT.NPUFFS(NS)) GO TO 33
251             IF (IQCOD(N,NG,NS).EQ.1) IQCOD(N,NG,NS)=0
252             IF (IQCOD(N,NG,NS).EQ.2) IQCOD(N,NG.NS)=4
253             GO TO 29
254           ENDIF
255     C
256     C               ***COMPUTE SIGMAY AND SIGMAZ.
257           CALL SIGGS(N,NG,NS,CLK,DELTAT,2)
258     C
259           ICONFG=3
260     C               ***COMPUTE CONC FOR QUADRILATERAL.
261           CALL CALC(N,NG,NS,ICONFG,NPLT,DELTAT,NGGGD)
262     C
263           IF (IQCOD(N,NG,NS).EQ.5)  THEN
264     C               ***LAST PUFF FOR TIME STEP.
265             ICONFG=2
266             CALL CALC(N,NG,NS,ICONFG,NPLT,DELTAT,NGGGD)
267             IQCOD(N,NG,NS)=1
268             GO TO 33
269           ENDIF
270           IF (IQCOD(N,NG,NS).EQ.2)  THEN
271             ICONFG=2
272             CALL UPDATE(N,NG,NS,ICONFG,DELTAT,NGGGD)
273             N=N+1
274             IF (N.GT.NPUFFS(NS)) GO TO 33
275             GO TO 29
276           ENDIF
277           GO TO 31
278     C
279        33 CONTINUE
280     C
281     C               ***WRITE OUT CONCENTRATION FIELDS FOR THIS TIME STEP.
282     C               ***CONCENTRATIONS STILL NEED CORRECTING TO STANDARD
283     C               ***CONDITIONS.
284           WRITE (LUO) CLK,NS
285           CLKI=CLK+DELTAT/2.
286           IF (NRGRID.NE.0)  THEN
287             DO 34 L=1,NPLT
288             DO 34 K=1,NRGRID
289             DO 34 J=1,NPTSY(K)
290             BI=(YREC(J,K)-RIY0)/SLY1
291             DO 44 II=1,NPTSX(K)
292               AI=(XREC(II,K)-RIX0)/SLX1
293               CALL BLPINT (AI,BI,CLKI,RMIX,ISTB,USTR,CAPL,WSTR,TSURF,PRESS)
294               CHI(II,J,K,L)=CHI(II,J,K,L)*STDF*TSURF/PRESS
295        44     CONTINUE
296               WRITE (LUO) (CHI(I,J,K,L),I=1,NPTSX(K))
297        34     CONTINUE
298             ENDIF
299             IF (NINDR.NE.0)  THEN
300               DO 35 L=1,NPLT
301               DO 54 II=1,NINDR
302                 AI=(XIREC(II)-RIX0)/SLX1
303                 BI=(YIREC(II)-RIY0)/SLY1
304                 CALL BLPINT (AI,BI,CLKI,RMIX,ISTB,USTR,CAPL,WSTR,TSURF,PRESS)
305                 CHIS(II,L)=CHIS(II,L)*STDF*TSURF/PRESS
306        54     CONTINUE
307               WRITE (LUO) (CHIS(I,L),I=1,NINDR)
308        35     CONTINUE
309             ENDIF
310             IF (NRGRID.NE.0)  THEN
```

```
311             DO 36 L=1,NPLT
312             DO 36 K=1,NRGRID
313             DO 36 J=1,NPTSY(K)
314             DO 36 I=1,NPTSX(K)
315               CHI(I,J,K,L)=0.
316          36   CONTINUE
317             ENDIF
318             IF (NINDR.NE.0)  THEN
319               DO 37 L=1,NPLT
320               DO 37 I=1,NINDR
321                 CHIS(I,L)=0.
322          37   CONTINUE
323             ENDIF
324       C
325       C               ***END OF LOOP ON SOURCES
326          38 CONTINUE
327       C
328       C               ***END OF PUFF RELEASE PER HOUR LOOP
329          90 CONTINUE
330       C
331             IF (NH.GT.NHOURS-2) GO TO 95
332             DO 40 L=1,NGAMA
333               DO 39 I=1,2
334               DO 39 J=1,NTERMS(I)
335                 FI(J,L,I)=FI(J,L,I+1)
336                 GI(J,L,I)=GI(J,L,I+1)
337                 GMF(J,L,I)=GMF(J,L,I+1)
338          39   CONTINUE
339               READ (12,END=96) ICC,THETTA(3),NTERMS(3),(FI(J.L.3),GI(J.L,3),
340            +                     GMF(J,L,3),J=1,NTERMS(3))
341          40 CONTINUE
342       C
343       C                ***READ IN ALL BOUNDARY LAYER PARAMETERS.
344             DO 42 L=1,NYCS
345               DO 41 J=1,NXCS
346               DO 41 I=1,2
347               DO 41 NB=1,7
348                 BLP(J,L,I,NB)=BLP(J,L,I+1,NB)
349          41   CONTINUE
350               READ (17) ICC,(BLP(J,L,3,1),J=1,NXCS)
351               READ (17) ICC,(IS(J),J=1,NXCS)
352               DO 45 J=1,NXCS
353                 BLP(J,L,3,2)=REAL(IS(J))
354          45   CONTINUE
355               READ (20) ICC,(BLP(J,L.3,3),J=1,NXCS)
356               READ (20) ICC,(BLP(J,L,3,4),J=1,NXCS)
357               READ (20) ICC,(BLP(J,L,3,5),J=1,NXCS)
358               READ (20) ICC,(BLP(J,L,3,6),J=1,NXCS)
359               READ (20) ICC,(BLP(J,L,3,7),J=1,NXCS)
360          42 CONTINUE
361       C
362       C               ***READ IN NEXT COUPLING COEFFICIENT
363             DO 75 L=1,NVALE
364               DO 74 J=1,NSEG(L)
365               DO 74 I=1,2
366                 COEF(J,L,I)=COEF(J,L,I+1)
367          74   CONTINUE
368               READ(27)ICC,IDV,(COEF(J,L,3),J=1,NSEG(L))
369          75 CONTINUE
370       C
371       C                ***END OF LOOP ON HOURS IN SIMULATION
372          95 CONTINUE
```

```
373            GO TO 98
374         96 CONTINUE
375            WRITE(6,97)
376      C
377         98 STOP
378         97 FORMAT (2X,'SIMULATION END TIME AFTER MET DATA SET END TIME.')
379            END
```

```
C  SUBROUTINE: POLUT.BLPINT
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO INTERPOLATE IN TIME AND SPACE
C  THE BOUNDARY LAYER PARAMETERS GIVEN THE LOCATION OF A POINT, AND
C  THE ARRAYS OF GRIDDED VALUES FOR THREE TIMES.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE BLPINT (XG,YG,CLK,HMIX,IS,USTR,CAPL,WSTR,TSURF,PRESS)
C
      PARAMETER
     +  NPG=10
C
      DIMENSION
     +  BLPI(2,7),          BLPA(7)
C
      COMMON /CLOCK1/
     +  CLK1,               CLK2,             CLK3
C
      COMMON /BLPAR1/
     +  BLP(NPG,NPG,3,7), XCS,            YCS,            HCELLX,
     +  HCELLY,             NXCS,           NYCS
C
C
C               ***CALCULATE ARRAY INDICIES.
      II=INT(XCS*XG-0.5)+1
      JJ=INT(YCS*YG-0.5)+1
C
C               ***CALCULATE WEIGHTING FACTORS.
      X2=XG*XCS-REAL(II)+0.5
      X1=1.-X2
      Y2=YG*YCS-REAL(JJ)+0.5
      Y1=1.-Y2
C
C               ***LOOP COMPUTING BOUNDARY LAYER PARAMETERS AT TWO TIMES
C               *** TO INTERPOLATE BETWEEN.
      NS=0
      NTS=0
      IF (CLK.GE.CLK2) NTS=1
      DO 9 NT=1+NTS,2+NTS
        NS=NS+1
C               ***CHECK TO SEE IF POINT IS ON HALF-CELL BORDERS OF DOMAIN
        IF (XG.LT.HCELLX) GO TO 1
        IF (YG.LT.HCELLY) GO TO 4
        IF (XG.GE.1.-HCELLX) GO TO 5
        IF (YG.GE.1.-HCELLY) GO TO 8
C               ***COMPUTE BOUNDARY LAYER PARAMETERS AT ALPHA AND BETA.
        DO 11 NB=1,7
          BLPI(NS,NB)=X1*Y1*BLP(II,JJ,NT,NB)+X2*Y1*BLP(II+1,JJ,NT,NB)+
     +                X1*Y2*BLP(II,JJ+1,NT,NB)+X2*Y2*BLP(II+1,JJ+1,NT,NB)
   11   CONTINUE
        GO TO 9
    1   II=1
          IF (YG.LT.HCELLY) GO TO 2
          IF (YG.GE.1.-HCELLY) GO TO 3
          DO 12 NB=1,7
            BLPI(NS,NB)=Y2*BLP(II,JJ+1,NT,NB)+Y1*BLP(II,JJ.NT,NB)
   12     CONTINUE
          GO TO 9
    2   JJ=1
          DO 13 NB=1,7
```

```
063                 BLPI(NS,NB)=BLP(II,JJ,NT,NB)
064          13     CONTINUE
065                 GO TO 9
066           3   JJ=NYCS
067                 DO 14 NB=1,7
068                   BLPI(NS,NB)=BLP(II,JJ,NT,NB)
069          14     CONTINUE
070                 GO TO 9
071           4   JJ=1
072                 DO 15 NB=1,7
073                   BLPI(NS,NB)=X2*BLP(II+1,JJ.NT,NB)+X1*BLP(II,JJ.NT,NB)
074          15     CONTINUE
075                 GO TO 9
076           5   II=NXCS    *
077                 IF (YG.LT.HCELLY) GO TO 6
078                 IF (YG.GE.1.0-HCELLY) GO TO 7
079                 DO 16 NB=1,7
080                   BLPI(NS,NB)=Y2*BLP(II,JJ+1,NT,NB)+Y1*BLP(II,JJ,NT,NB)
081          16     CONTINUE
082                 GO TO 9
083           6   JJ=1
084                 DO 17 NB=1,7
085                   BLPI(NS,NB)=BLP(II,JJ.NT,NB)
086          17     CONTINUE
087                 GO TO 9
088           7   JJ=NYCS
089                 DO 18 NB=1,7
090                   BLPI(NS,NB)=BLP(II,JJ,NT,NB)
091          18     CONTINUE
092                 GO TO 9
093           8   JJ=NYCS
094                 DO 19 NB=1,7
095                   BLPI(NS,NB)=X2*BLP(II+1,JJ,NT,NB)+X1*BLP(II,JJ.NT,NB)
096          19     CONTINUE
097           9 CONTINUE
098     C
099     C                ***DETERMINE TIME INTERPOLATED MIXING HEIGHT.
100               T2=(CLK-CLK1)/(CLK2-CLK1)
101               IF (CLK.GE.CLK2) T2=ABS((CLK-CLK2)/(CLK3-CLK2))
102               T1=1.-T2
103               DO 20 NB=1,7
104                 BLPA(NB)=T2*BLPI(2,NB)+T1*BLPI(1,NB)
105          20   CONTINUE
106               HMIX=BLPA(1)
107               IS=NINT(BLPA(2))
108               USTR=BLPA(3)
109               CAPL=BLPA(4)
110               WSTR=BLPA(5)
111               TSURF=BLPA(6)
112               PRESS=BLPA(7)
113     C
114               RETURN
115               END
```

```
C  SUBROUTINE POLUT.CALC
C  APRIL 1985
C
C    THIS SUBROUTINE IS USED TO DEFINE SEGMENTS OR QUADRILATERALS,
C  DETERMINE IF THEY SHOULD BE DIVIDED, COMPUTE GROUND-LEVEL
C  CONCENTRATIONS AT NEARBY RECEPTORS, AND UPDATE THE PUFF
C  ARRAYS CHARACTERISTICS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE CALC(N,NG,NS,ICONFG,NPLT,DELTAT,NSGA)
C
      PARAMETER
     +   NPU=100,      NSG=12,      NSC=5
C
      LOGICAL
     +   DIVD
C
      COMMON /PUFF1/
     +   A(NPU,NSG,NSC),      B(NPU,NSG,NSC),      IPCOD(NPU,NSG,NSC),
     +   SGPRY(NPU,NSC),      SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
     +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
     +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
     +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),      ZMIX(NPU,NSG,NSC),
     +   QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
     +   NPUFFV(NSC)
C
      COMMON /TPDT/
     +    ANQ(2),         BNQ(2),         DSMQ(2),        HTMIXQ(2),
     +    IPCODQ(2),      SYDPQ(2),       SYDGQ(2),       SYSHQ(2),
     +    SYTOQ(2),       SZDQ(2),        SZTOQ(2),       ZNQ(2),
     +    ANQS(2),        BNQS(2),        DSMQS(2),       ZNQS(2),
     +    AF(4),          BF(4),          QF(2)
C
      DO 5 JJ=1,4
        AF(JJ)=0.
        BF(JJ)=0.
    5 CONTINUE
C
      K=N
      IF(ICONFG.EQ.3) K=N-1
      DO 10 L=1,NPLT
        QF(L)=QTP(K,NG,NS,L)
   10 CONTINUE
C
      CALL DEFINE (N,NG,NS,ICONFG,NPLT)
      CALL DIVCK (ICONFG,DIVD,NPLT)
      IF (DIVD)  THEN
        CALL DIVIDE (ICONFG,NPLT)
        CALL CONS (NS,ICONFG,NPLT)
      ENDIF
      CALL UPDATE (N,NG,NS,ICONFG,DELTAT,NSGA)
C
      RETURN
      END
```

```
C  SUBROUTINE POLUT.CALCVA
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO DETERMINE THE NUMBER OF 'POINT
C  SOURCES' EACH VALLEY SEGMENT SHOULD BE REPRESENTED AS.  IT ALSO
C  CALLS THE SUBROUTINES TO DIVIDE PUFF QUADRILATERALS AND COMPUTE
C  GROND-LEVEL CONCENTRATIONS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE CALCVA (N,NG,NS,ICONFG,NPLT,NSGT,DELTAT)
C
      PARAMETER
     +    NPU=100,       NSC=5,         NSG=12,        IP=50
C
      LOGICAL
     +    DIVD
C
      DIMENSION
     +    WF1(IP),       WF2(IP),       WF3(IP),       WF4(IP),
     +    WF(IP,4),      NPT(4),        DA(4),         DB(4),
     +    X(4),          Y(4),          XS(4),         YS(4)
C
      EQUIVALENCE
     +    (WF1(1),WF(1,1)),    (WF2(1),WF(1,2)),
     +    (WF3(1),WF(1,3)),    (WF4(1),WF(1,4))
C
      COMMON /MAIN2/
     +    HA,            HH,            IHH,           IX0,
     +    IY0,           ROA,           RXY,           SLX,
     +    SLY
C
      COMMON /PUFF1/
     +   A(NPU,NSG,NSC),     B(NPU,NSG,NSC),     IPCOD(NPU,NSG,NSC),
     +   SGPRY(NPU,NSC),     SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
     +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
     +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG.NSC),
     +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),     ZMIX(NPU,NSG.NSC),
     +   QTP(NPU.NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
     +   NPUFFV(NSC)
C
      COMMON /TPDT/
     +    ANQ(2),        BNQ(2),        DSMQ(2),       HTMIXQ(2),
     +    IPCODQ(2),     SYDPQ(2),      SYDGQ(2),      SYSHQ(2),
     +    SYTOQ(2),      SZDQ(2),       SZTOQ(2),      ZNQ(2),
     +    ANQS(2),       BNQS(2),       DSMQS(2),      ZNQS(2),
     +    AF(4),         BF(4),         QF(2)
C
C
      RIX0=REAL(IX0)
      RIY0=REAL(IY0)
      IF (ICONFG.EQ.3)  THEN
        K=N-1
        J=2
        I=1
      ELSE
        K=N
        J=ICONFG
        I=ICONFG
      ENDIF
C
      NPT(1)=0
```

```
063           NPT(4)=0
064           IF(ICONFG.EQ.3) THEN
065             IF (N.EQ.1) GO TO 1
066             DA(1)=A(N-1,NG+1,NS)-A(N-1,NG,NS)
067             DB(1)=B(N-1,NG+1,NS)-B(N-1,NG,NS)
068             X(1)=(A(N-1,NG,NS)*SLX+RIX0)
069             Y(1)=(B(N-1,NG,NS)*SLY+RIY0)
070             XS(1)=(A(N-1,NG+1,NS)*SLX+RIX0)
071             YS(1)=(B(N-1,NG+1,NS)*SLY+RIY0)
072             SEGL=SQRT((XS(1)-X(1))**2+(YS(1)-Y(1))**2)*1000.
073             CALL WEIGHT(SEGL,SGTOY(N-1,NG,NS),SGTOY(N-1,NG,NS),NPT(1),WF1)
074         1   CONTINUE
075             DA(4)=ANQS(I)-ANQ(I)
076             DB(4)=BNQS(I)-BNQ(I)
077             X(4)=(ANQ(I)*SLX+RIX0)
078             Y(4)=(BNQ(I)*SLY+RIY0)
079             XS(4)=(ANQS(I)*SLX+RIX0)
080             YS(4)=(BNQS(I)*SLY+RIY0)
081             SEGL=SQRT((XS(4)-X(4))**2+(YS(4)-Y(4))**2)*1000.
082             CALL WEIGHT(SEGL,SYTOQ(I),SYTOQ(I),NPT(4),WF4)
083           ENDIF
084     C
085           DA(2)=A(N,NG+1,NS)-A(N,NG,NS)
086           DB(2)=B(N,NG+1,NS)-B(N,NG,NS)
087           X(2)=(A(N,NG,NS)*SLX+RIX0)
088           Y(2)=(B(N,NG,NS)*SLY+RIY0)
089           XS(2)=(A(N,NG+1,NS)*SLX+RIX0)
090           YS(2)=(B(N,NG+1,NS)*SLY+RIY0)
091           SEGL=SQRT((XS(2)-X(2))**2+(YS(2)-Y(2))**2)*1000.
092           CALL WEIGHT(SEGL,SGTOY(N,NG,NS),SGTOY(N,NG,NS),NPT(2),WF2)
093     C
094           DA(3)=ANQS(J)-ANQ(J)
095           DB(3)=BNQS(J)-BNQ(J)
096           X(3)=(ANQ(J)*SLX+RIX0)
097           Y(3)=(BNQ(J)*SLY+RIY0)
098           XS(3)=(ANQS(J)*SLX+RIX0)
099           YS(3)=(BNQS(J)*SLY+RIY0)
100           SEGL=SQRT((XS(3)-X(3))**2+(YS(3)-Y(3))**2)*1000.
101           CALL WEIGHT(SEGL,SYTOQ(J),SYTOQ(J),NPT(3),WF3)
102     C
103           NP=MAX(NPT(1),NPT(2),NPT(3),NPT(4))
104           DO 10 J1=1,4
105             IF(NP.EQ.NPT(J1)) IX=J1
106        10 CONTINUE
107     C
108           DO 11 J1=1,NP
109             DO 12 L=1,NPLT
110               QF(L)=QTP(K,NG,NS,L)*WF(J1,IX)
111        12   CONTINUE
112             JJ=J1-1
113             IF(J1.EQ.1) THEN
114               DO 4 K1=1,4
115                 AF(K1)=0.
116                 BF(K1)=0.
117          4    CONTINUE
118             ELSE
119               DO 5 K1=1,4
120                 AF(K1)=DA(K1)*WF(JJ,IX)+AF(K1)
121                 BF(K1)=DB(K1)*WF(JJ,IX)+BF(K1)
122          5    CONTINUE
123             ENDIF
124             CALL DEFINE (N,NG,NS,ICONFG,NPLT)
```

```
125             CALL DIVCK (ICONFG,DIVD,NPLT)
126             IF (DIVD)  THEN
127               CALL DIVIDE (ICONFG,NPLT)
128               CALL CONS (NS,ICONFG,NPLT)
129             ENDIF
130          11 CONTINUE
131             CALL UPDATE (N,NG,NS,ICONFG,DELTAT,NSGT)
132      C
133             RETURN
134             END
```

```
001     C  SUBROUTINE POLUT.CLEAN
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO ELIMINATE UNUSED PUFFS FROM THE PUFF
005     C  CHARACTERSTICS ARRAYS.
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE CLEAN(NS)
010     C
011           PARAMETER
012          +   NPU=100,        NSC=5,        NSG=12
013     C
014           COMMON /PUFF1/
015          +   A(NPU,NSG,NSC),      B(NPU,NSG,NSC),      IPCOD(NPU,NSG,NSC),
016          +   SGPRY(NPU,NSC),      SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
017          +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
018          +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
019          +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),      ZMIX(NPU,NSG,NSC),
020          +   QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
021          +   NPUFFV(NSC)
022     C
023           NPUFO=NPUFFS(NS)
024           IP=0
025           NG=1
026           DO 10 N=1,NPUFO
027           IF(IQCOD(N,NG,NS).NE.4) THEN
028             NC=N-IP
029             A(NC,NG,NS)=A(N,NG,NS)
030             B(NC,NG,NS)=B(N,NG,NS)
031             IPCOD(NC,NG,NS)=IPCOD(N,NG,NS)
032             SGPRY(NC,NS)=SGPRY(N,NS)
033             SGDPY(NC,NG,NS)=SGDPY(N,NG,NS)
034             SGDGY(NC,NG,NS)=SGDGY(N,NG,NS)
035             SGSHY(NC,NG,NS)=SGSHY(N,NG,NS)
036             SGTOY(NC,NG,NS)=SGTOY(N,NG,NS)
037             SGPRZ(NC,NS)=SGPRZ(N,NS)
038             SGDZ(NC,NG,NS)=SGDZ(N,NG,NS)
039             SGTOZ(NC,NG,NS)=SGTOZ(N,NG,NS)
040             STOTM(NC,NG,NS)=STOTM(N,NG,NS)
041             TIME(NC,NG,NS)=TIME(N,NG,NS)
042             Z(NC,NG,NS)=Z(N,NG,NS)
043             ZMIX(NC,NG,NS)=ZMIX(N,NG,NS)
044             QTP(NC,NG,NS,1)=QTP(N,NG,NS,1)
045             QTP(NC,NG,NS,2)=QTP(N,NG,NS,2)
046             IQCOD(NC,NG,NS)=IQCOD(N,NG,NS)
047           ELSE
048             IP=IP+1
049           ENDIF
050        10 CONTINUE
051     C
052           IF(IP.EQ.0) THEN
053             WRITE(6,100)
054             STOP 'CLEAN'
055           ENDIF
056     C
057           NPUFN=NPUFO-IP
058           NPUFFS(NS)=NPUFN
059     C
060           RETURN
061       100 FORMAT(1H ,' PUFF ARRAYS ARE FULL -- MUST RUN POLUT AGAIN FOR',
062          +       /,1X,' LESS TIME OR LOWER PUFF RELEASE RATE.')
```

```
063          END
```

```
C  SUBROUTINE POLUT.CLEANV
C  APRIL 1985
C
C    THIS SUBROUTINE IS USED TO ELIMINATE UNUSED PUFFS FROM THE PUFF
C  CHARACTERISTICS ARRAYS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE CLEANV(NS,NSGT)
C
      PARAMETER
     +   NPU=100,        NSC=5,        NSG=12
C
      LOGICAL
     +   ALL
C
      COMMON /PUFF1/
     +   A(NPU,NSG,NSC),      B(NPU,NSG,NSC),      IPCOD(NPU,NSG,NSC),
     +   SGPRY(NPU,NSC),      SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
     +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
     +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
     +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),      ZMIX(NPU,NSG,NSC),
     +   QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
     +   NPUFFV(NSC)
C
      NPUFO=NPUFFV(NS)
      IP=0
      DO 10 N=1,NPUFO
        ALL=.TRUE.
        DO 2 NG=2,NSGT+1
          IF(IQCOD(N,NG,NS).NE.4) ALL=.FALSE.
    2   CONTINUE
        IF(ALL) THEN
          NC=N-IP
          DO 5 NG=2,NSGT+2
            A(NC,NG,NS)=A(N,NG,NS)
            B(NC,NG,NS)=B(N,NG,NS)
            IPCOD(NC,NG,NS)=IPCOD(N,NG,NS)
            SGDPY(NC,NG,NS)=SGDPY(N,NG,NS)
            SGDGY(NC,NG,NS)=SGDGY(N,NG,NS)
            SGSHY(NC,NG,NS)=SGSHY(N,NG,NS)
            SGTOY(NC,NG,NS)=SGTOY(N,NG,NS)
            SGDZ(NC,NG,NS)=SGDZ(N,NG,NS)
            SGTOZ(NC,NG,NS)=SGTOZ(N,NG,NS)
            STOTM(NC,NG,NS)=STOTM(N,NG,NS)
            TIME(NC,NG,NS)=TIME(N,NG,NS)
            Z(NC,NG,NS)=Z(N,NG,NS)
            ZMIX(NC,NG,NS)=ZMIX(N,NG,NS)
            QTP(NC,NG,NS,1)=QTP(N,NG,NS,1)
            QTP(NC,NG,NS,2)=QTP(N,NG,NS,2)
            IQCOD(NC,NG,NS)=IQCOD(N,NG,NS)
    5     CONTINUE
        ELSE
          IP=IP+1
        ENDIF
   10 CONTINUE
C
      IF(IP.EQ.0) THEN
        WRITE(6,100)
        STOP 'CLEANV'
      ENDIF
C
```

```
063           NPUFN=NPUFO-IP
064           NPUFFV(NS)=NPUFN
065     C
066           RETURN
067       100 FORMAT(1H ,' PUFF ARRAYS ARE FULL -- MUST RUN POLUT AGAIN FOR',
068          +       /,1X,' LESS TIME OR LOWER PUFF RELEASE RATE.')
069           END
```

```
001     C  SUBROUTINE POLUT.CONS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE GROUND-LEVEL POLLUTANT
005     C  CONCENTRATIONS.
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE CONS (NS,ICONFG,NPLT)
010     C
011           PARAMETER
012          +    IP=50,          NSC=5
013     C
014           COMMON /QUAOUT/
015          +    IPC(IP,IP),     ND,             NT,             SY(IP,IP),
016          +    SZ(IP,IP),      XX(IP,IP),      YY(IP,IP),      ZM(IP,IP),
017          +    ZZ(IP,IP),      ZT(IP,IP),      QMP(IP,IP,2)
018     C
019           COMMON /REC/
020          +    NINDR,          NPTSX(4),       NPTSY(4),       NRGRID,
021          +    RSPAC(4),       STAX(4),        STAY(4),        XIREC(10),
022          +    XREC(25,4),     YIREC(10),      YREC(25,4),     ZIREC(10),
023          +    ZREC(25,25,4)
024     C
025           COMMON /NOC1/
026          +    NGBPT(NSC),     NIBPT(NSC),     IGBAD(100,NSC),IIBAD(10,NSC)
027     C
028           COMMON /CONC1/
029          +    CHI(25,25,4,2),CHIS(10,2)
030     C
031           TPI=6.283185
032           TPI15=TPI**1.5
033           MICGR=10.**6.
034     C
035     C              ***COMPUTE CONCENTRATIONS AT RECEPTORS FOR EACH
036     C              ***RECTANGULAR GRID.
037           IF (NRGRID.EQ.0) GO TO 5
038           DO 4 NG=1,NRGRID
039             IC1=NG*10000
040             DO 3 J=1,NPTSY(NG)
041             IC2=J*100
042             DO 3 I=1,NPTSX(NG)
043             IC=IC1+IC2+I
044             DO 1 IW=1,NGBPT(NS)
045               IF(IGBAD(IW,NS).EQ.IC) GO TO 3
046         1   CONTINUE
047             DO 3 JJ=2,NT
048               IF (ICONFG.EQ.3)  THEN
049                 ISTART=2
050                 ISTOP=ND
051               ELSE
052                 ISTART=1
053                 ISTOP=1
054               ENDIF
055               JN=JJ+1
056               IF(JJ.EQ.NT) JN=JJ-1
057             DO 3 II=ISTART,ISTOP
058               CALL LITLR(XREC(I,NG),YREC(J,NG),ZREC(I,J,NG),XX(II,JN),
059          +          YY(II,JN),ZT(II,JN),XX(II,JJ),YY(II,JJ),ZT(II,JJ),DIST)
060               PUFRAD=5.*SY(II,JJ)
061               IF (DIST.GT.PUFRAD) GO TO 3
062               HDF=EXP(-.5*AMIN1(30.,((DIST/SY(II,JJ))**2)))
```

```
063               IF (IPC(II,JJ).EQ.1) THEN
064                 CALL HTLOC(XREC(I,NG),YREC(J,NG),ZREC(I,J,NG),XX(II,JN),
065          +           YY(II,JN),ZT(II,JN),XX(II,JJ),YY(II,JJ),ZT(II,JJ),
066          +           ZZ(II,JJ),DAGL)
067                 CON1=1./(SY(II,JJ)**2*SZ(II,JJ)*TPI15)
068                 CALL VERTDF (SZ(II,JJ),DAGL,ZM(II,JJ),VDF)
069                 CHI1=(CON1*VDF*HDF)
070               ENDIF
071               IF (IPC(II,JJ).EQ.2)  THEN
072                 CON1=1./(SY(II,JJ)**2*ZM(II,JJ)*TPI)
073                 VDF=1.
074                 CHI1=(CON1*VDF*HDF)
075               ENDIF
076               DO 2 L=1,NPLT
077                 CHI(I,J,NG,L)=CHI(I,J,NG,L)+CHI1*QMP(II,JJ,L)*MICGR
078         2     CONTINUE
079         3   CONTINUE
080         4 CONTINUE
081     C
082     C              ***COMPUTE CONCENTRATIONS AT THE INDIVIDUALLY SPECIFIED
083     C              ***RECEPTORS.
084         5 IF (NINDR.EQ.0) RETURN
085           DO 7 J=1,NINDR
086           DO 8 IW=1,NIBPT(NS)
087             IF(IIBAD(IW,NS).EQ.J) GO TO 7
088         8 CONTINUE
089           DO 7 JJ=2,NT
090             IF (ICONFG.EQ.3)  THEN
091               ISTART=2
092               ISTOP=ND
093             ELSE
094               ISTART=1
095               ISTOP=1
096             ENDIF
097             JN=JJ+1
098             IF(JJ.EQ.NT) JN=JJ-1
099           DO 7 II=ISTART,ISTOP
100             CALL LITLR(XIREC(J),YIREC(J),ZIREC(J),XX(II,JN),YY(II,JN),
101          +             ZT(II,JN),XX(II,JJ),YY(II,JJ),ZT(II,JJ),DIST)
102             PUFRAD=5.*SY(II,JJ)
103             IF (DIST.GT.PUFRAD) GO TO 7
104             HDF=EXP(-.5*AMIN1(30.,((DIST/SY(II,JJ))**2)))
105             IF (IPC(II,JJ).EQ.1) THEN
106               CALL HTLOC(XIREC(J),YIREC(J),ZIREC(J),XX(II,JN),
107          +         YY(II,JN),ZT(II,JN),XX(II,JJ),YY(II,JJ),ZT(II,JJ),
108          +         ZZ(II,JJ),DAGL)
109               CON1=1./(SY(II,JJ)**2*SZ(II,JJ)*TPI15)
110               CALL VERTDF (SZ(II,JJ),DAGL,ZM(II,JJ),VDF)
111               CHI1=(CON1*VDF*HDF)
112             ENDIF
113             IF (IPC(II,JJ).EQ.2)  THEN
114               CON1=1./(SY(II,JJ)**2*ZM(II,JJ)*TPI)
115               VDF=1.
116               CHI1=(CON1*VDF*HDF)
117             ENDIF
118             DO 6 L=1,NPLT
119               CHIS(J,L)=CHIS(J,L)+CHI1*QMP(II,JJ,L)*MICGR
120         6   CONTINUE
121         7 CONTINUE
122     C
123           RETURN
124           END
```

```
001     C  SUBROUTINE POLUT.DEFINE
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO SPECIFY VALUES OF VARIABLES
005     C  USED AS INPUT TO THE PUFF SPLITTING SUBROUTINE DIVIDE.
006     C
007     C    FOR A SEGMENT THE '2' AND '3' POSITIONS IN THE ARRAYS ARE USED.
008     C  THE '2' POSITION REPRESENTS THE PUFF AT TIME 'T', AND THE '3'
009     C  POSITION REPRESENTS THE PUFF AT TIME 'T+DELTA T'.
010     C
011     C    THE INPUTS ARE:
012     C      ICONFG - 1, 2, OR 3 AND SPECIFIES THE CONFIGURATION OF THE
013     C               PUFFS.
014     C             1 - SEGMENT (TWO PUFFS) AT OLDER PUFF POSITION 1.
015     C             2 - SEGMENT AT NEWER PUFF POSITION 2.
016     C             3 - FOUR PUFFS FORMING A QUADRILATERAL.
017     C      N - PUFF NUMBER.
018     C      NG - VALLEY SEGMENT NUMBER.
019     C      NS - STATION NUMBER.
020     C      NPLT - NUMBER OF POLLUTANTS.
021     C
022     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
023     C
024           SUBROUTINE DEFINE (N,NG,NS,ICONFG,NPLT)
025     C
026           PARAMETER
027          +    NPU=100,        NSC=5,          NSG=12
028     C
029           COMMON /MAIN2/
030          +    HA,             HH,             IHH,            IX0,
031          +    IY0,            ROA,            RXY,            SLX,
032          +    SLY
033     C
034           COMMON /QUAIN/
035          +    IPCO(4),        SGY(4),         SGZ(4),         X(4),
036          +    Y(4),           Z3(4),          ZMIH(4),        AP(4),
037          +    BP(4),          QP(2)
038     C
039           COMMON /PUFF1/
040          +   A(NPU,NSG,NSC),      B(NPU,NSG,NSC),      IPCOD(NPU,NSG,NSC),
041          +   SGPRY(NPU,NSC),      SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
042          +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
043          +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
044          +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),      ZMIX(NPU,NSG,NSC),
045          +   QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
046          +   NPUFFV(NSC)
047     C
048           COMMON /TPDT/
049          +    ANQ(2),         BNQ(2),         DSMQ(2),        HTMIXQ(2),
050          +    IPCODQ(2),      SYDPQ(2),       SYDGQ(2),       SYSHQ(2),
051          +    SYTOQ(2),       SZDQ(2),        SZTOQ(2),       ZNQ(2),
052          +    ANQS(2),        BNQS(2),        DSMQS(2),       ZNQS(2),
053          +    AF(4),          BF(4),          QF(2)
054     C
055           RIX0=REAL(IX0)
056           RIY0=REAL(IY0)
057           IF (ICONFG.EQ.3)  THEN
058             K=N-1
059             J=2
060             I=1
061           ELSE
062             K=N
```

```
063             J=ICONFG
064             I=ICONFG
065           ENDIF
066     C
067           DO 10 L=1,NPLT
068             QP(L)=QF(L)
069        10 CONTINUE
070     C
071           IF (N.EQ.1) GO TO 1
072           AP(1)=A(N-1,NG,NS)+AF(1)
073           BP(1)=B(N-1,NG,NS)+BF(1)
074           X(1)=(AP(1)*SLX+RIX0)*1000.
075           Y(1)=(BP(1)*SLY+RIY0)*1000.
076           Z3(1)=Z(N-1,NG,NS)
077           SGY(1)=SGTOY(N-1,NG,NS)
078           SGZ(1)=SGTOZ(N-1,NG,NS)
079           ZMIH(1)=ZMIX(N-1,NG,NS)
080           IPCO(1)=IPCOD(N-1,NG,NS)
081         1 CONTINUE
082     C
083           AP(2)=A(N,NG,NS)+AF(2)
084           BP(2)=B(N,NG,NS)+BF(2)
085           X(2)=(AP(2)*SLX+RIX0)*1000.
086           Y(2)=(BP(2)*SLY+RIY0)*1000.
087           Z3(2)=Z(N,NG,NS)
088           SGY(2)=SGTOY(N,NG,NS)
089           SGZ(2)=SGTOZ(N,NG,NS)
090           ZMIH(2)=ZMIX(N,NG,NS)
091           IPCO(2)=IPCOD(N,NG,NS)
092     C
093           AP(3)=ANQ(J)+AF(3)
094           BP(3)=BNQ(J)+BF(3)
095           X(3)=(AP(3)*SLX+RIX0)*1000.
096           Y(3)=(BP(3)*SLY+RIY0)*1000.
097           Z3(3)=ZNQ(J)
098           SGY(3)=SYTOQ(J)
099           SGZ(3)=SZTOQ(J)
100           ZMIH(3)=HTMIXQ(J)
101           IPCO(3)=IPCODQ(J)
102     C
103           AP(4)=ANQ(I)+AF(4)
104           BP(4)=BNQ(I)+BF(4)
105           X(4)=(AP(4)*SLX+RIX0)*1000.
106           Y(4)=(BP(4)*SLY+RIY0)*1000.
107           Z3(4)=ZNQ(I)
108           SGY(4)=SYTOQ(I)
109           SGZ(4)=SZTOQ(I)
110           ZMIH(4)=HTMIXQ(I)
111           IPCO(4)=IPCODQ(I)
112     C
113           RETURN
114           END
```

```
C  SUBROUTINE POLUT.DIVCK
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE IS USED TO CHECK WHETHER OR NOT SUBROUTINES
C  DIVIDE AND CONS SHOULD BE CALLED.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE DIVCK (ICONFG,DIVD,NPLT)
C
      LOGICAL
     +    DIVD
C
      COMMON /QUAIN/
     +    IPCO(4),         SGY(4),        SGZ(4),        X(4),
     +    Y(4),            Z3(4),         ZMIH(4),       AP(4),
     +    BP(4),           QP(2)
C
      COMMON /RGRIDC/
     +    CXG(4),          CYG(4),        RADG(4)
C
      COMMON /REC/
     +    NINDR,           NPTSX(4),      NPTSY(4),      NRGRID,
     +    RSPAC(4),        STAX(4),       STAY(4),       XIREC(10),
     +    XREC(25,4),      YIREC(10),     YREC(25,4),    ZIREC(10),
     +    ZREC(25,25,4)
C
      DIVD=.FALSE.
      IF (ICONFG.EQ.3)  THEN
        ISTA=1
        ISTP=4
      ELSE
        ISTA=2
        ISTP=3
      ENDIF
C
C             ***CHECK IF POLLUTANT MASS OF PRIMARY PUFF IS ZERO.
      DO 5 L=1,NPLT
        IF(QP(L).NE.0.) GO TO 6
    5 CONTINUE
      RETURN
C
C             ***CHECK TO SEE IF EACH RECEPTOR GRID IS WITHIN 5 SIGMA'S
C             ***OF THE PUFFS.
    6 IF (NRGRID.EQ.0) GO TO 2
      DO 1 K=ISTA,ISTP
        L=K
        J=K+1
        IF (J.EQ.5) THEN
          J=4
          L=1
        ENDIF
        IF (ICONFG.EQ.3) THEN
          SEGL=SQRT((X(J)-X(L))**2+(Y(J)-Y(L))**2)
        ELSE
          SEGL=SQRT((X(3)-X(2))**2+(Y(3)-Y(2))**2)
        ENDIF
        PUFRAD=5.*SGY(K)
        DISTP=AMAX1(PUFRAD,SEGL)
      DO 1 I=1,NRGRID
        DISTR=SQRT((X(K)-CXG(I))**2+(Y(K)-CYG(I))**2)
        IF (DISTP.GT.DISTR-RADG(I)) THEN
```

```
063                 DIVD=.TRUE.
064                 RETURN
065               ENDIF
066           1 CONTINUE
067     C
068     C                 ***CHECK TO SEE IF EACH INDIV
069     C                 ***SIGMA'S OF THE PUFFS.
070           2 CONTINUE
071             IF (NINDR.EQ.0) GO TO 4
072             DO 3 K=ISTA,ISTP
073               L=K
074               J=K+1
075               IF (J.EQ.5) THEN
076                 J=4
077                 L=1
078               ENDIF
079               IF (ICONFG.EQ.3) THEN
080                 SEGL=SQRT((X(J)-X(L))**2+(Y(J)-Y(
081               ELSE
082                 SEGL=SQRT((X(3)-X(2))**2+(Y(3)-Y(
083               ENDIF
084               PUFRAD=5.*SGY(K)
085               DISTP=AMAX1(PUFRAD,SEGL)
086             DO 3 I=1,NINDR
087               DISTR=SQRT((X(K)-XIREC(I))**2+(Y(K)
088               IF (DISTP.GT.DISTR) THEN
089                 DIVD=.TRUE.
090                 RETURN
091               ENDIF
092           3 CONTINUE
093     C
094           4 RETURN
095             END
```

```
F EACH INDIVIDUAL RECEPTOR IS WITHIN 5
 PUFFS.


**2+(Y(J)-Y(L))**2)

**2+(Y(3)-Y(2))**2)

)

))**2+(Y(K)-YIREC(I))**2)
```

```
C  SUBROUTINE POLUT.DIVIDE
C  DATE: APRIL 1985
C
C    GIVEN THE LOCATION AND SIGMA Y'S OF FOUR CORNER PUFFS OF
C  A QUADRILATERAL, THIS SUBROUTINE COMPUTES THE LOCATIONS OF ALL
C  PUFFS WITHIN THE QUADRILATERAL SUCH THAT THE PUFFS ARE
C  SEPARATED BY NOT MORE THAN ONE SIGMA Y.
C
C  THE NOMENCLATURE IS:
C
C                               (3)
C                         3 +------------+ 4  PLUME AT T+DT
C                           I            I
C                       (2) I            I (4)
C                           I            I
C                         2 +----------- + 1  PLUME AT TIME T
C                               (1)
C
C    THE NUMBERS IN PARENTHESES DENOTE SEGMENT IDENTIFICATIONS.
C    THE + DENOTES THE MARKING PUFF CENTERS.
C    THE PLUME SEGMENT DEFINED BY PUFFS '1' AND '2' MOVES TO
C    THE POSITION DEFINED BY PUFFS '3' AND '4' OVER TIME STEP 'DT'.
C
C    THE POSITION OF ALL THE PUFFS WITHIN THE QUADRILATERAL ARE
C  STORED IN TWO DIMENSIONAL ARRAYS WHERE THE FIRST INDEX RANGES
C  FROM 1 TO ND AND THE SECOND INDEX RANGES FROM 1 TO NT.  'ND'
C  IS THE NUMBER OF PUFFS ON SEGMENTS (1) AND (3), AND 'NT' IS
C  THE NUMBER OF PUFFS ON SEGMENTS (2) AND (4).  POSITION 1,1 IS
C  PUFF '2'.
C
C    THE INPUTS ARE:
C      X - ARRAY OF X-COORDINATES OF FOUR CORNER PUFFS (M).
C      Y - ARRAY OF Y-COORDINATES OF FOUR CORNER PUFFS (M).
C      Z - ARRAY OF Z-COORDINATES OF FOUR CORNER PUFFS (M AGL).
C      SGY - ARRAY OF SIGMA Y'S OF FOUR PUFFS (M).
C      SGZ - ARRAY OF SIGMA Z'S OF FOUR PUFFS (M).
C      ZMIH - HEIGHT OF MIXING DEPTH AT PUFF LOCATION (M AGL).
C      IPCO - VERTICAL DISTRIBUTION CODE ( 1, 2, OR 3).
C
C    THE OUTPUTS ARE:
C      XX - ARRAY OF X-COORDINATES OF ALL PUFFS INCLUDING ORIGINAL
C           FOUR PUFFS (M).
C      YY - ARRAY OF Y-COORDINATES.
C      ZZ - ARRAY OF Z-COORDINATES (M AGL).
C      ZM - HEIGHT OF MIXING DEPTH AT PUFF LOCATIONS (M AGL).
C      IPC - VERTICAL DISTRIBUTION CODE ( 1, 2, OR 3).
C      ND - NUMBER OF PUFFS ON SEGMENTS '1' AND '3'.
C      NT - NUMBER OF PUFFS ON SEGMENTS '2' AND '4'.
C
C      THE SIGMAS AT ALL INTERMEDIATE PUFFS ARE DETERMINED USING
C      NON-LINEAR INTERPOLATION BETWEEN THE FOUR CORNER PUFFS.
C      THE HEIGHT ABOVE GROUND LEVEL OF ALL INTERMEDIATE PUFFS
C      IS DETERMINED BY LINEARLY INTERPOLATING THE HEIGHT ABOVE
C      GROUND OF THE FOUR CORNER PUFFS.
C   THE MIXING DEPTH AT ALL INTERMEDIATE PUFFS IS DETERMINED
C   BY LINEARLY INTERPOLATING THE DEPTH AT THE FOUR CORNER PUFFS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE DIVIDE (ICONFG,NPLT)
C
      PARAMETER
```

```
063           +    IP=50
064     C
065            DIMENSION
066           +    SEGL(4),        ZMM(4),         ZTH(4),         A(IP,2),
067           +    B(IP,2),        WFS(IP),        WFT(IP)
068     C
069            COMMON /QUAIN/
070           +    IPCO(4),        SGY(4),         SGZ(4),         X(4),
071           +    Y(4),           Z3(4),          ZMIH(4),        AP(4),
072           +    BP(4),          QP(2)
073     C
074            COMMON /QUAOUT/
075           +    IPC(IP,IP),     ND,             NT,             SY(IP,IP),
076           +    SZ(IP,IP),      XX(IP,IP),      YY(IP,IP),      ZM(IP,IP),
077           +    ZZ(IP,IP),      ZT(IP,IP),      QMP(IP,IP,2)
078     C
079     C
080            IF (ICONFG.NE.3)  THEN
081     C             ***DIVIDE PLUME SEGMENT INTO NUMBER OF PUFFS TO GET
082     C             ***SUFFICIENT OVERLAP.
083            CALL TETRI (AP(3),BP(3),ZTH(3),DZA,DZB)
084            ZMM(3)=Z3(3)+ZTH(3)
085            CALL TETRI (AP(2),BP(2),ZTH(2),DZA,DZB)
086            ZMM(2)=Z3(2)+ZTH(2)
087            SEGL(2)=SQRT((X(3)-X(2))**2+(Y(3)-Y(2))**2+(ZMM(3)-ZMM(2))**2)
088            CALL WEIGHT(SEGL(2),SGY(2),SGY(3),NSG,WFS)
089            NPUT=NSG+1
090            JHALF=NPUT/2
091            RNPUT=REAL(NPUT)
092            A(1,1)=AP(2)
093            B(1,1)=BP(2)
094            XX(1,1)=X(2)
095            YY(1,1)=Y(2)
096            ZZ(1,1)=Z3(2)
097            ZM(1,1)=ZMIH(2)
098            ZT(1,1)=ZTH(2)
099            IPC(1,1)=IPCO(2)
100            SY(1,1)=SGY(2)
101            SZ(1,1)=SGZ(2)
102            IF (NPUT.EQ.2) GO TO 13
103            DX=X(3)-X(2)
104            DY=Y(3)-Y(2)
105            DA=AP(3)-AP(2)
106            DB=BP(3)-BP(2)
107            DZ=Z3(3)-Z3(2)
108            DMH=ZMIH(3)-ZMIH(2)
109            DSY=SGY(3)-SGY(2)
110            DSZ=SGZ(3)-SGZ(2)
111            DO 1 J=1,NPUT-1
112              JJ=J+1
113              NJ=NPUT-J
114              XX(1,JJ)=XX(1,J)+DX*WFS(NJ)
115              YY(1,JJ)=YY(1,J)+DY*WFS(NJ)
116              ZZ(1,JJ)=ZZ(1,J)+DZ*WFS(NJ)
117              ZM(1,JJ)=ZM(1,J)+DMH*WFS(NJ)
118              SY(1,JJ)=SY(1,J)+DSY*WFS(NJ)
119              SZ(1,JJ)=SZ(1,J)+DSZ*WFS(NJ)
120              A(1,1)=A(1,1)+DA*WFS(NJ)
121              B(1,1)=B(1,1)+DB*WFS(NJ)
122              CALL TETRI(A(1,1),B(1,1),ZT(1,JJ),DZA,DZB)
123              IF (IPCO(2).EQ.IPCO(3))  THEN
124                IPC(1,JJ)=IPCO(2)
```

```
125             ELSE
126               IF (J.LE.JHALF) IPC(1,JJ)=IPCO(2)
127               IF (J.GT.JHALF) IPC(1,JJ)=IPCO(3)
128             ENDIF
129           1 CONTINUE
130          13 CONTINUE
131             NT=NPUT
132             XX(1,NT)=X(3)
133             YY(1,NT)=Y(3)
134             ZZ(1,NT)=Z3(3)
135             ZM(1,NT)=ZMIH(3)
136             ZT(1,NT)=ZTH(3)
137             IPC(1,NT)=IPCO(3)
138             SY(1,NT)=SGY(3)
139             SZ(1,NT)=SGZ(3)
140             ND=1
141     C
142     C                **DETERMINE POLLUTANT MASS ASSOCIATED WITH EACH PUFF.
143           DO 20 L=1,NPLT
144           DO 20 J=2,NPUT
145             NJ=NPUT-(J-1)
146             QMP(1,J,L)=QP(L)*WFS(NJ)
147          20 CONTINUE
148             RETURN
149             ENDIF
150     C
151     C                ***DIVIDE THE QUADRILATERAL INTO PUFFS.
152     C
153     C                ***COMPUTE HEIGHT MSL OF 4 PUFFS.
154           DO 2 J=1,4
155             CALL TETRI (AP(J),BP(J),ZTH(J),DZA,DZB)
156             ZMM(J)=Z3(J)+ZTH(J)
157           2 CONTINUE
158     C
159     C                ***COMPUTE THE LENGTHS OF THE FOUR SEGMENTS.
160           DO 3 K=1,4
161             L=K
162             J=K+1
163             IF (J.EQ.5)  THEN
164               J=4
165               L=1
166             ENDIF
167             SEGL(K)=SQRT((X(J)-X(L))**2+(Y(J)-Y(L))**2+(ZMM(J)-ZMM(L))**2)
168             IF (SEGL(K).LE.0.)  THEN
169               WRITE (6,10)
170               STOP 'DIVIDE'
171             ENDIF
172           3 CONTINUE
173     C
174     C                ***COMPUTE NUMBER OF PUFFS ON 'S' SEGMENTS.
175           CALL WEIGHT(SEGL(1),SGY(2),SGY(1),NSG1,WFS)
176           CALL WEIGHT(SEGL(3),SGY(3),SGY(4),NSG2,WFS)
177           IF(NSG1.GE.NSG2) THEN
178             CALL WEIGHT(SEGL(1),SGY(2),SGY(1),NSG,WFS)
179           ELSE
180             CALL WEIGHT(SEGL(3),SGY(3),SGY(4),NSG,WFS)
181           ENDIF
182           NPS=NSG+1
183     C                ***COMPUTE NUMBER OF PUFFS ON 'T' SEGMENT PERPENDICULAR
184     C                ***TO 'S' SEGMENT
185           DOT21=(X(3)-X(2))*(X(1)-X(2))+(Y(3)-Y(2))*(Y(1)-Y(2))+
186          +      (ZMM(3)-ZMM(2))*(ZMM(1)-ZMM(2))
```

```
187           DOT41=(X(4)-X(1))*(X(2)-X(1))+(Y(4)-Y(1))*(Y(2)-Y(1))+
188          +       (ZMM(4)-ZMM(1))*(ZMM(2)-ZMM(1))
189           AARG2=DOT21/(SEGL(1)*SEGL(2))
190           ARG2=ABS(AARG2)
191           IF(ARG2.GT.1.0001.OR.ARG2.LT..9999)THEN
192             PTD2=SEGL(2)*(SIN(ACOS(ARG2)))
193           ELSE
194             PTD2=0.
195           ENDIF
196           AARG4=DOT41/(SEGL(1)*SEGL(4))
197           ARG4=ABS(AARG4)
198           IF(ARG4.GT.1.0001.OR.ARG4.LT..9999)THEN
199             PTD4=SEGL(4)*(SIN(ACOS(ARG4)))
200           ELSE
201             PTD4=0.
202           ENDIF
203           CALL WEIGHT(PTD2,SGY(2),SGY(3),NSG1,WFT)
204           CALL WEIGHT(PTD4,SGY(1),SGY(4),NSG2,WFT)
205           IF(NSG1.GE.NSG2) THEN
206             CALL WEIGHT(PTD2,SGY(2),SGY(3),NSG,WFT)
207           ELSE
208             CALL WEIGHT(PTD4,SGY(1),SGY(4),NSG,WFT)
209           ENDIF
210           NPUT=NSG+1
211     C
212     C               ***DIVIDE TWO 'S' SEGMENTS (1 AND 3) INTO NUMBER OF PUFFS
213     C               ***COMPUTING PUFF LOCATIONS, SIGMA'S MIXING HEIGHTS AND
214     C               ***VERTICAL DISTRIBUTION CODES.
215           IHALF=NPS/2
216           DO 5 KK=1,2
217             IF (KK.EQ.1)  THEN
218               NT=1
219               L=2
220               M=1
221             ELSE
222               NT=NPUT
223               L=3
224               M=4
225             ENDIF
226             A(1,KK)=AP(L)
227             B(1,KK)=BP(L)
228             XX(1,NT)=X(L)
229             YY(1,NT)=Y(L)
230             ZZ(1,NT)=Z3(L)
231             ZM(1,NT)=ZMIH(L)
232             ZT(1,NT)=ZTH(L)
233             IPC(1,NT)=IPCO(L)
234             SY(1,NT)=SGY(L)
235             SZ(1,NT)=SGZ(L)
236             IF (NPS.EQ.2) GO TO 14
237             DX=X(M)-X(L)
238             DY=Y(M)-Y(L)
239             DA=AP(M)-AP(L)
240             DB=BP(M)-BP(L)
241             DZ=Z3(M)-Z3(L)
242             DMH=ZMIH(M)-ZMIH(L)
243             DSY=SGY(M)-SGY(L)
244             DSZ=SGZ(M)-SGZ(L)
245             DO 4 I=1,NPS-1
246               II=I+1
247               NI=NPS-I
248               XX(II,NT)=XX(I,NT)+DX*WFS(NI)
```

```
249               YY(II,NT)=YY(I,NT)+DY*WFS(NI)
250               ZZ(II,NT)=ZZ(I,NT)+DZ*WFS(NI)
251               ZM(II,NT)=ZM(I,NT)+DMH*WFS(NI)
252               SY(II,NT)=SY(I,NT)+DSY*WFS(NI)
253               SZ(II,NT)=SZ(I,NT)+DSZ*WFS(NI)
254               A(II,KK)=A(I,KK)+DA*WFS(NI)
255               B(II,KK)=B(I,KK)+DB*WFS(NI)
256               CALL TETRI(A(II,KK),B(II,KK),ZT(II,NT),DZA,DZB)
257               IF (IPCO(L).EQ.IPCO(M))  THEN
258                 IPC(II,NT)=IPCO(L)
259               ELSE
260                 IF (I.LE.IHALF) IPC(II,NT)=IPCO(L)
261                 IF (I.GT.IHALF) IPC(II,NT)=IPCO(M)
262               ENDIF
263          4    CONTINUE
264         14    CONTINUE
265               ND=NPS
266               A(ND,KK)=AP(M)
267               B(ND,KK)=BP(M)
268               XX(ND,NT)=X(M)
269               YY(ND,NT)=Y(M)
270               ZZ(ND,NT)=Z3(M)
271               ZM(ND,NT)=ZMIH(M)
272               ZT(ND,NT)=ZTH(M)
273               IPC(ND,NT)=IPCO(M)
274               SY(ND,NT)=SGY(M)
275               SZ(ND,NT)=SGZ(M)
276          5 CONTINUE
277     C
278     C              **COMPUTE POLLUTANT MASS ASSOCIATED WITH EACH PUFF ON
279     C              **THE S SEGMENT
280           DO 22 L=1,NPLT
281           DO 22 I=2,NPS
282             NI=NPS-(I-1)
283             QMP(I,1,L)=QP(L)*WFS(NI)
284         22 CONTINUE
285     C
286     C              ***COMPUTE LOCATIONS AND SIGMA'S OF ALL PUFFS.
287           NT=NPUT
288           IF (NT.EQ.2) GO TO 8
289           JHALF=NT/2
290           DO 7 I=1,NPS
291             DX=XX(I,NT)-XX(I,1)
292             DY=YY(I,NT)-YY(I,1)
293             DA=A(I,2)-A(I,1)
294             DB=B(I,2)-B(I,1)
295             DZ=ZZ(I,NT)-ZZ(I,1)
296             DMH=ZM(I,NT)-ZM(I,1)
297             DSY=SY(I,NT)-SY(I,1)
298             DSZ=SZ(I,NT)-SZ(I,1)
299             DO 6 J=1,NPUT-1
300               JJ=J+1
301               NJ=NPUT-J
302               XX(I,JJ)=XX(I,J)+DX*WFT(NJ)
303               YY(I,JJ)=YY(I,J)+DY*WFT(NJ)
304               ZZ(I,JJ)=ZZ(I,J)+DZ*WFT(NJ)
305               ZM(I,JJ)=ZM(I,J)+DMH*WFT(NJ)
306               SY(I,JJ)=SY(I,J)+DSY*WFT(NJ)
307               SZ(I,JJ)=SZ(I,J)+DSZ*WFT(NJ)
308               A(I,1)=A(I,1)+DA*WFT(NJ)
309               B(I,1)=B(I,1)+DB*WFT(NJ)
310               CALL TETRI(A(I,1),B(I,1),ZT(I,JJ),DZA,DZB)
```

```
311              IF (IPC(I,1).EQ.IPC(I,NPUT))  THEN
312                IPC(I,JJ)=IPC(I,1)
313              ELSE
314                IF (J.LE.JHALF) IPC(I,JJ)=IPC(I,1)
315                IF (J.GT.JHALF) IPC(I,JJ)=IPC(I,NPUT)
316              ENDIF
317         6   CONTINUE
318         7 CONTINUE
319     C               **COMPUTE POLLUTANT MASS ASSOCIATED WITH EACH PUFF.
320         8 DO 25 L=1,NPLT
321           DO 25 J=2,NPUT
322           NJ=NPUT-(J-1)
323           DO 25 I=1,NPS
324             QMP(I,J,L)=QMP(I,1,L)*WFT(NJ)
325        25 CONTINUE
326     C
327     C               ***NUMBER OF PUFFS ON 'S' SEGMENTS (1 AND 3) = ND.
328     C               ***NUMBER OF PUFFS ON 'T' SEGMENTS (2 AND 4) = NT.
329           ND=NPS
330           NT=NPUT
331     C
332           RETURN
333        10 FORMAT(5X,' SEGMENT LENGTH LESS THAN OR EQUAL TO ZERO.')
334           END
```

```
001     C  SUBROUTINE:  POLUT.HTLOC
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO COMPUTE THE 'LOCAL' PUFF HEIGHT ABOVE
005     C  THE GROUND-LEVEL RECEPTOR.
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C     XR,YR,ZR = COORDINATES OF RECEPTOR (M)
009     C     XP,YP,ZP = COORDINATES OF PUFF (M)
010     C     XPO,YPO,ZPO = COORDINATES OF PUFF CENTER PROJECTED VERTICALLY
011     C                   DOWN TO GROUND-LEVEL (M)
012     C     H = HEIGHT OF PUFF ABOVE GROUND (M)
013     C (THE VERTICAL COORDINATES ARE IN M MSL)
014     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
015     C  THE OUTPUTS ARE:
016     C     HL = LOCAL PUFF HEIGHT ABOVE THE GROUND-LEVEL RECEPTOR (M)
017     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
018     C
019           SUBROUTINE HTLOC(XR,YR,ZR,XP,YP,ZP,XPO,YPO,ZPO,H,HL)
020     C
021           AZ=ZP-ZPO
022           AX=(XP-XPO)
023           AY=(YP-YPO)
024           DD=AX**2+AY**2
025           E=-AX*AZ/DD
026           F=-AY*AZ/DD
027           E2=E**2
028           F2=F**2
029           G=1.
030           G2=1.
031           SUM=E2+F2+G2
032           RK=(E*(XP-XR)+F*(YP-YR)+G*(ZP-ZR))/SUM
033           RK2=RK**2
034           D=SQRT(RK2*SUM)
035           IF(RK.GT.0.) D=-D
036           HL=H-D
037     C
038           RETURN
039           END
```

```
001     C  SUBROUTINE POLUT.INIT
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE READS IN ALL THE INITIAL INFORMATION REQUIRED BY
005     C  POLUT.
006     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
007     C
008           SUBROUTINE INIT(ISTART,ISTOPT,IPRR,NPLT,LUO)
009     C
010           PARAMETER
011          +    NGA=9,          NPO=21,         NPT=50,         NVA=5,
012          +    NSC=5,          NPG=10,         NSG=12
013     C
014           CHARACTER*10
015          +    NAMEG,          NAMER,          NAMES,          IDRUN,
016          +    NAMVAL(10),     NASVAL,         USENAM
017     C
018           CHARACTER*3
019          +    PLTNAM
020     C
021           LOGICAL
022          +    LUFORM,         WOFF,           VALL
023     C
024           DIMENSION
025          +    IS(NPG)
026     C
027           COMMON /HTADJ/
028          +    CT(6)
029     C
030           COMMON /MAIN2/
031          +    HA,             HH,             IHH,            IXO,
032          +    IYO,            ROA,            RXY,            SLX,
033          +    SLY
034     C
035           COMMON /TER1/
036          +    ICST,           NXCT,           NYCT,           RGHX(NPT,NPT),
037          +    RGHY(NPT,NPT), THS(NPT,NPT),   XCT,            YCT,
038          +    DXT,            DYT,            HDXT,           HDYT
039     C
040           COMMON /WINDS1/
041          +    FI(NPO,NGA,3), GI(NPO,NGA,3), GMF(NPO,NGA,3), HTGAMA(NGA),
042          +    THETTA(3),      NGAMA,          NTERMS(3)
043     C
044           COMMON /BLPAR1/
045          +  BLP(NPG,NPG,3,7), XCS,           YCS,            HCELLX,
046          +  HCELLY,           NXCS,          NYCS
047     C
048           COMMON /VALLY1/
049          +   COEF(NSG,NVA,3),COXV(NVA),      COYV(NVA),      WSV(NVA),
050          +   SEGX(NSG,NVA),  SEGY(NSG,NVA), NSEG(NVA),      IDVAL(NVA),
051          +   NVALE,          QM(NSG,NSC,2), VSEGL(NSG,NVA),WIDV(NSG,NVA)
052     C
053           COMMON /REC/
054          +    NINDR,          NPTSX(4),       NPTSY(4),       NRGRID,
055          +    RSPAC(4),       STAX(4),        STAY(4),        XIREC(10),
056          +    XREC(25,4),     YIREC(10),      YREC(25,4),     ZIREC(10),
057          +    ZREC(25,25,4)
058     C
059           COMMON /RGRIDC/
060          +    CXG(4),         CYG(4),         RADG(4)
061     C
062           COMMON /RUNSP/
```

```
063          +    LUFORM,         METHOD,         WOFF
064     C
065           COMMON /SOURC1/
066          +    EMIS(NSC,2),    NSOURC,         RS(NSC),        TS(NSC),
067          +    VS(NSC),        XSL(NSC),       YSL(NSC),       ZSH(NSC),
068          +    ELS(NSC),       AS(NSC),        BS(NSC),        SIGIN(NSC),
069          +    IDSVAL(NSC),    VALL(NSC),      INDXV(NSC)
070     C
071           COMMON /NAME1/
072          +    NAMEG(4),       NAMER(10),      NAMES(NSC),     PLTNAM(2),
073          +    NASVAL(NSC)
074     C
075     C
076 .   C              ***READ IN RUN INITIALIZATION INFORMATION.
077           LUO=14
078           READ (13,*) IDRUN,USENAM,IBTZ
079           READ (13,*) IBEG,IEND,IPRR,LUFORM,WOFF,METHOD
080           READ (13,*) IX0,IY0,SLX,SLY
081           READ (13,*) (CT(I),I=1,6)
082           RIX0=REAL(IX0)
083           RIY0=REAL(IY0)
084           RXY=SLX/SLY
085     C
086     C              ***READ IN SMOOTHED TERRAIN HEIGHT AND ROUGHNESS DATA
087           READ (11) IX0T,IY0T,NXCT,NYCT,ICST
088           SLXT=REAL(NXCT*ICST)
089           SLYT=REAL(NYCT*ICST)
090           XCT=REAL(NXCT)
091           YCT=REAL(NYCT)
092           DXT=1.0/XCT
093           DYT=1.0/YCT
094           HDXT=0.5*DXT
095           HDYT=0.5*DYT
096           IF(IX0T.NE.IX0.OR.IY0T.NE.IY0.OR.SLXT.NE.SLX.OR.SLYT.NE.SLY) THEN
097             WRITE (6,47)
098             STOP 'INIT'
099           ENDIF
100           DO 15 M=1,NYCT
101             READ (11) (THS(L,M),L=1,NXCT)
102        15 CONTINUE
103           DO 70 M=1,NYCT
104             READ (11) (RGHX(L,M),L=1,NXCT)
105             READ (11) (RGHY(L,M),L=1,NXCT)
106        70 CONTINUE
107     C
108     C              ***READ IN RECEPTOR SPECIFICATIONS
109           READ (15,*) NRGRID,NINDR
110           IF (NRGRID.NE.0)  THEN
111             DO 4 K=1,NRGRID
112               READ (15,*) NAMEG(K),XU,NPTSX(K),YU,NPTSY(K),RSPAC(K),ICOD
113               GO TO (2,1,1,3), ICOD
114         1     IZONE=ICOD
115               CALL UTM (IZONE,XU,YU,STAX(K),STAY(K))
116               GO TO 4
117         2     CALL LAT (XU,YU,STAX(K),STAY(K))
118               GO TO 4
119         3     STAX(K)=XU
120               STAY(K)=YU
121         4   CONTINUE
122           ENDIF
123           IF (NINDR.NE.0)  THEN
124             DO 8 K=1,NINDR
```

```
125               READ (15,*) NAMER(K),XU,YU,ICOD
126               GO TO (6,5,5,7), ICOD
127          5    IZONE=ICOD
128               CALL UTM (IZONE,XU,YU,XIREC(K),YIREC(K))
129               GO TO 8
130          6    CALL LAT (XU,YU,XIREC(K),YIREC(K))
131               GO TO 8
132          7    XIREC(K)=XU
133               YIREC(K)=YU
134          8   CONTINUE
135             ENDIF
136     C
137     C              ***READ IN SOURCE INFORMATION
138             READ (16,*) NSOURC,NPLT,(PLTNAM(J),J=1,NPLT)
139             DO 14 J=1,NSOURC
140               READ (16,*) NAMES(J),XU,YU,ICOD,VALL(J)
141               IF(VALL(J)) READ (16,*) IDSVAL(J),NASVAL(J)
142               READ (16,*) SIGIN(J),ZSH(J),VS(J),TS(J),RS(J)
143               READ (16,*) (EMIS(J,I),I=1,NPLT)
144               GO TO (10,9,9,11), ICOD
145          9    IZONE=ICOD
146               CALL UTM (IZONE,XU,YU,XSL(J),YSL(J))
147               GO TO 12
148         10    CALL LAT (XU,YU,XSL(J),YSL(J))
149               GO TO 12
150         11    XSL(J)=XU
151               YSL(J)=YU
152         12    CONTINUE
153               IF (XSL(J).LT.RIX0.OR.XSL(J).GT.SLX+RIX0.OR.YSL(J).LT.RIY0.OR.
154           +       YSL(J).GT.SLY+RIY0)  THEN
155                 WRITE (6,45) J
156                 STOP 'INIT'
157               ENDIF
158             AS(J)=(XSL(J)-RIX0)/SLX
159             BS(J)=(YSL(J)-RIY0)/SLY
160             CALL TETRI(AS(J),BS(J),ELS(J),DZA,DZB)
161         14  CONTINUE
162     C
163     C              ***SET UP SIMULATION CLOCK.
164             CALL JULIAN (IBEG,7,7,IHB,IJB)
165             ISTART=(IJB-1)*24+IHB
166             CALL JULIAN (IEND,7,7,IHE,IJE)
167             ISTOPT=(IJE-1)*24+IHE
168     C
169     C              ***READ IN AMPLITUDE FUNCTIONS FOR FIRST THREE HOURS
170             READ (12) IX0W,IY0W,SLXW,SLYW
171             IF(IX0W.NE.IX0.OR.IY0W.NE.IY0.OR.SLXW.NE.SLX.OR.SLYW.NE.SLY) THEN
172               WRITE (6,49)
173               STOP 'INIT'
174             ENDIF
175             READ (12) NGAMA,(HTGAMA(KK),KK=1,NGAMA),IHH,HH
176     C                  **READ TO STARTING HOUR
177         31  DO 32 L=1,NGAMA
178               READ (12,END=33) ICC,THETTA(1),NTERMS(1),(FI(J,L,1),GI(J,L,1),
179           +                    GMF(J,L,1),J=1,NTERMS(1))
180         32  CONTINUE
181             IF(ICC.NE.ISTART) GO TO 31
182     C                **READ IN NEXT TWO HOURS OF DATA
183             DO 16 IT=2,3
184             DO 16 L=1,NGAMA
185               READ (12) ICC,THETTA(IT),NTERMS(IT),(FI(J,L,IT),GI(J,L,IT),
186           +             GMF(J,L,IT),J=1,NTERMS(IT))
```

```
   16 CONTINUE
C
C              ***READ IN STABILITY AND MIXING HEIGHT FOR FIRST THREE
C              ***HOURS.
        READ (17) IXOS,IYOS,NXCS,NYCS,ICSS
        SLXS=REAL(NXCS*ICSS)
        SLYS=REAL(NYCS*ICSS)
        XCS=REAL(NXCS)
        YCS=REAL(NYCS)
        HCELLX=1./(2.*XCS)
        HCELLY=1./(2.*YCS)
        IF(IXOS.NE.IX0.OR.IYOS.NE.IY0.OR.SLXS.NE.SLX.OR.SLYS.NE.SLY) THEN
          WRITE (6,51)
          STOP 'INIT'
        ENDIF
C                **READ TO STARTING HOUR
   34 DO 36 J=1,NYCS
          READ (17) ICC,(BLP(I,J,1,1),I=1,NXCS)
          READ (17) ICC,(IS(I),I=1,NXCS)
          DO 35 I=1,NXCS
            BLP(I,J,1,2)=REAL(IS(I))
   35     CONTINUE
   36 CONTINUE
        IF(ICC.NE.ISTART) GO TO 34
C                **READ IN NEXT TWO HOURS
        DO 17 IT=2,3
        DO 17 J=1,NYCS
          READ (17) ICC,(BLP(I,J,IT,1),I=1,NXCS)
          READ (17) ICC,(IS(I),I=1,NXCS)
          DO 30 I=1,NXCS
            BLP(I,J,IT,2)=REAL(IS(I))
   30     CONTINUE
   17 CONTINUE
C
C                ***READ IN TEMPERATURE, PRESSURE, FRICTION VELOCITY,
C                ***CONVECTIVE VELOCITY, AND MONIN-OBUKOV LENGTH FOR
C                ***FIRST THREE HOURS
        READ (20) IXOS,IYOS,NXCS,NYCS,ICSS
        SLXS=REAL(NXCS*ICSS)
        SLYS=REAL(NYCS*ICSS)
        IF(IXOS.NE.IX0.OR.IYOS.NE.IY0.OR.SLXS.NE.SLX.OR.SLYS.NE.SLY) THEN
          WRITE (6,53)
          STOP 'INIT'
        ENDIF
C                **READ TO STARTING HOUR
   37 DO 38 J=1,NYCS
          READ(20) ICC,(BLP(I,J,1,3),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,1,4),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,1,5),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,1,6),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,1,7),I=1,NXCS)
   38 CONTINUE
        IF(ICC.NE.ISTART) GO TO 37
C                **READ IN NEXT TWO HOURS
        DO 71 IT=2,3
        DO 71 J=1,NYCS
          READ(20) ICC,(BLP(I,J,IT,3),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,IT,4),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,IT,5),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,IT,6),I=1,NXCS)
          READ(20) ICC,(BLP(I,J,IT,7),I=1,NXCS)
   71 CONTINUE
```

```
249     C
250     C               ***READ IN VALLEY COUPLING COEFFICIENTS
251           READ(27) NVALE
252           DO 72 NV=1,NVALE
253             READ(27) IDVAL(NV),NAMVAL(NV),NSEG(NV),COXV(NV),COYV(NV),
254          +  WSV(NV),(SEGX(NSE,NV),SEGY(NSE,NV),WIDV(NSE,NV),NSE=1,NSEG(NV))
255        72 CONTINUE
256     C                **READ TO STARTING HOUR
257        39 DO 40 NV=1,NVALE
258             READ(27)ICC,IDV,(COEF(NSE,NV,1),NSE=1,NSEG(NV))
259        40 CONTINUE
260           IF(ICC.NE.ISTART) GO TO 39
261     C                **READ IN NEXT TWO HOURS
262           DO 73 IT=2,3
263           DO 73 NV=1,NVALE
264             READ(27)ICC,IDV,(COEF(NSE,NV,IT),NSE=1,NSEG(NV))
265        73 CONTINUE
266     C
267     C               ***DETERMINE INDEX OF VALLEYS THAT SOURCES LOCATED IN.
268           DO 80 NS=1,NSOURC
269             INDXV(NS)=-1
270             IF(.NOT.VALL(NS)) GO TO 80
271             DO 79 NV=1,NVALE
272               IF(IDVAL(NV).EQ.IDSVAL(NS)) INDXV(NS)=NV
273        79   CONTINUE
274             IF(INDXV(NS).LE.0) THEN
275               WRITE(6,77) NAMES(NS)
276               STOP 'INIT'
277             ENDIF
278        80 CONTINUE
279     C
280     C               ***CALCULATE VALLEY SEGMENT LENGTHS
281           DO 82 NS=1,NSOURC
282             IF(VALL(NS)) THEN
283               NV=INDXV(NS)
284               DO 81 NG=1,NSEG(NV)
285                 IF(NG.LT.NSEG(NV)) THEN
286                   VSEGL(NG,NV)=SQRT((SEGX(NG+1,NV)-SEGX(NG,NV))**2+
287          +                          (SEGY(NG+1,NV)-SEGY(NG,NV))**2)
288                 ELSE
289                   VSEGL(NG,NV)=SQRT((XSL(NS)-SEGX(NG,NV))**2+
290          +                          (YSL(NS)-SEGY(NG,NV))**2)
291                 ENDIF
292        81       CONTINUE
293             ENDIF
294        82 CONTINUE
295     C
296     C               ***WRITE OUT RUN SPECIFICATION DATA TO UNIT 14.
297           WRITE (LUO) IDRUN,IBEG,IEND,IPRR
298           WRITE (LUO) IXO,IYO,SLX,SLY
299           WRITE (LUO) NRGRID,NINDR
300           IF (NRGRID.NE.0)  THEN
301           DO 18 K=1,NRGRID
302           WRITE (LUO)NAMEG(K),STAX(K),NPTSX(K),STAY(K),NPTSY(K),RSPAC(K)
303        18 CONTINUE
304           ENDIF
305           IF (NINDR.NE.0)  THEN
306           DO 19 K=1,NINDR
307           WRITE (LUO)NAMER(K),XIREC(K),YIREC(K)
308        19 CONTINUE
309           ENDIF
310           WRITE (LUO) NSOURC
```

```
311           DO 20 J=1,NSOURC
312           WRITE (LUO) NAMES(J),XSL(J),YSL(J)
313           WRITE (LUO) NPLT,(PLTNAM(I),I=1,NPLT)
314        20 CONTINUE
315     C
316     C                ***COMPUTE CENTER AND RADIUS OF RECTANGULAR RECEPTOR
317     C                ***GRIDS.
318           DO 21 M=1,NRGRID
319             XINC=RSPAC(M)*(REAL(NPTSX(M)-1))/2.*1000.
320             YINC=RSPAC(M)*(REAL(NPTSY(M)-1))/2.*1000.
321             CXG(M)=STAX(M)*1000.+XINC
322             CYG(M)=STAY(M)*1000.+YINC
323             RADG(M)=SQRT(XINC**2.+YINC**2.)
324        21 CONTINUE
325     C
326     C                ***COMPUTE COORDINATES OF RECEPTORS.
327           IF (NRGRID.NE.0)  THEN
328             DO 24 NG=1,NRGRID
329               DREC=RSPAC(NG)*1000.
330               DO 22 J=1,NPTSY(NG)
331                 YREC(J,NG)=STAY(NG)*1000.+REAL(J-1)*DREC
332        22       CONTINUE
333               DO 23 I=1,NPTSX(NG)
334                 XREC(I,NG)=STAX(NG)*1000.+REAL(I-1)*DREC
335        23       CONTINUE
336             DO 24 J=1,NPTSY(NG)
337               BET=(YREC(J,NG)/1000.-RIY0)/SLY
338             DO 24 I=1,NPTSX(NG)
339               ALP=(XREC(I,NG)/1000.-RIX0)/SLX
340               CALL TETRI (ALP,BET,ZREC(I,J,NG),DZX,DZY)
341        24   CONTINUE
342           ENDIF
343     C
344           IF (NINDR.NE.0)  THEN
345             DO 25 J=1,NINDR
346               BET=(YIREC(J)-RIY0)/SLY
347               ALP=(XIREC(J)-RIX0)/SLX
348               CALL TETRI (ALP,BET,ZIREC(J),DZX,DZY)
349               YIREC(J)=YIREC(J)*1000.
350               XIREC(J)=XIREC(J)*1000.
351        25   CONTINUE
352           ENDIF
353     C
354     C                ***CALCULATE RECEPTORS WITHIN MINIMUM DISTANCE TO REC.
355           CALL NOCALC
356     C                ***WRITE OUT SUMMARY TABLE
357           CALL OUT(IBEG,IEND,IJB,IJE,IHB,IHE,IPRR,NPLT,IDRUN,USENAM)
358           GO TO 75
359        33 WRITE(6,76)
360           STOP 'INIT'
361     C
362        75 RETURN
363        45 FORMAT (5X,' SOURCE',I4,'  IS NOT IN MODELING DOMAIN.')
364        47 FORMAT (2X,'DOMAIN SPECS SPECIFIED IN TERRAIN FILE NOT SAME AS','
365          +SPECIFIED IN RUN.')
366        49 FORMAT (2X,'DOMAIN SPECS SPECIFIED IN WINDS FILE NOT SAME AS',' SP
367          +ECIFIED IN RUN.')
368        51 FORMAT (2X,'DOMAIN SPECS FROM STABILITY FILE NOT SAME AS',' SPECIF
369          +IED IN RUN.')
370        53 FORMAT (2X,'DOMAIN SPECS FROM MIXING HEIGHT FILE NOT SAME AS',' SP
371          +ECIFIED IN RUN.')
372        76 FORMAT (2X,' REQUESTED SIMULATION START TIME IS NOT INCLUDED IN',
```

```
     +              ' MET DATA PERIOD.')
  77 FORMAT (2X,' VALLEY SPECS FOR SOURCE ',A10,' NOT INCLUDED IN',
     +              ' COEFF INPUT FILE.')
     END
```

```
001     C  SUBROUTINE: POLUT.JULIAN
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE JULIAN DATE AND HOUR-OF-
005     C  DAY IN THE BASE TIME ZONE GIVEN AN 8 DIGET INTEGER NUMBER
006     C  CODED WITH THE MONTH, DAY, YEAR, AND HOUR, AND THE TIME ZONE
007     C  OF THE CODED TIME.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       ITIME = 8 DIGET INTEGER NUMBER CODED WITH 'MO'DA'YR'HR'
011     C       ITZO = TIME ZONE OF CODED TIME = 0(ZULU), 5(EASTERN),
012     C                          6(CENTRAL), 7(MOUNTAIN), 8(PACIFIC).
013     C       IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL),
014     C                               7(MOUNTAIN), 8(PACIFIC).
015     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
016     C  THE OUTPUTS ARE:
017     C       IHR = HOUR-OF-DAY (01 TO 24)
018     C       JDATE = JULIAN DATE (1 TO 366)
019     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
020     C
021           SUBROUTINE JULIAN (ITIME,ITZO,IBTZ,IHR,JDATE)
022     C
023           DIMENSION
024          +    NDAY(12)
025     C
026           DATA NDAY/0,31,59,90,120,151,181,212,243,273,304.334/
027     C
028           TIM=REAL(ITIME)
029           MO=INT(TIM/10.**6.)
030           IDAY=INT(TIM/10.**4.)-MO*10**2
031           IYR=INT(TIM/10.**2.)-MO*10**4-IDAY*10**2
032           IHR=ITIME-INT(TIM/10.**2.)*10**2
033     C
034     C               ***CHECK FOR MONTH INDEX IN RANGE.
035           IF (MO.LE.0.OR.MO.GE.13)  THEN
036             WRITE (6,1)
037             STOP 'JULIAN'
038           ENDIF
039     C
040     C               ***CONVERT TO JULIAN DATE.
041           JDATE=IDAY+NDAY(MO)
042     C
043     C               ***ADJUST FOR LEAP YEAR.
044           A=FLOAT(IYR)/4-IYR/4
045           IF (A.EQ.0..AND.MO.GE.3) JDATE=JDATE+1
046     C
047     C               ***CONVERT TO BASE TIME ZONE
048           IHR=IHR+(ITZO-IBTZ)
049           IF(IHR.LE.0) THEN
050             IHR=IHR+24
051             JDATE=JDATE-1
052           ENDIF
053           IF(IHR.GT.24) THEN
054             IHR=IHR-24
055             JDATE=JDATE+1
056           ENDIF
057     C
058           RETURN
059         1 FORMAT (5X,'MONTH INDEX IS OUT OF RANGE.')
060           END
```

```
001     C  SUBROUTINE: POLUT.LAT
002     C  DATE: APRIL  1985
003     C
004     C    THIS SUBROUTINE IS USED TO CONVERT FROM LATITUDE-LONGITUDE
005     C  IN DEGREES TO GRAMA COORDINATES IN KILOMETERS.  EQUATIONS FOR
006     C  A TRANSVERSE MERCATOR PROJECTION ASSUMING THE EARTH IS SHAPED
007     C  AS A SPHERE ARE USED.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C       XLAT = LATITUDE OF POINT (DEG)
011     C       YLONG = LONGITUDE OF POINT (DEG)
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C       COX = X-COORDINATE IN GRAMA COORDINATE SYSTEM (KM)
015     C       COY = Y-COORDINATE IN GRAMA COORDINTAE SYSTEM (KM)
016     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
017     C  CONSTANT DEFINITIONS:
018     C       OX = 250.5  [DIST. OF GRAMA ORIGIN WEST OF CM (KM)]
019     C       OY = 4224.697  [DIST. IF GRAMA ORIGIN NORTH OF EQUATOR (KM)]
020     C       RAD = 6367.3858  [MEAN RADIUS OF EARTH (KM)]
021     C       C = .9996  [PROJECTION SCALING FACTOR]
022     C       CM = 109.  [CENTRAL MERIDIAN OF TRANS. MERCATOR PROJ. (DEG)]
023     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
024     C
025           SUBROUTINE LAT (XLAT,YLONG,COX,COY)
026     C
027           DOUBLE PRECISION A,B
028     C
029     C                ***DEFINE CONSTANT PARAMETERS.
030           CO1=3.141592654/180.
031           CO2=3.141592654/2.
032           RAD=6367.3858
033           CM=109.*CO1
034           OX=250.5
035           OY=4224.697
036           C=.9996
037     C
038     C                ***COMPUTE X COORINATE.
039           XLATR=XLAT*CO1
040           YLONGR=YLONG*CO1
041           A=COS(XLATR)*SIN(CM-YLONGR)
042           COX=.5*C*RAD*DLOG((1.+A)/(1.-A))+OX
043     C
044     C                ***COMPUTE Y COORDINATE.
045           B=1./TAN(XLATR)*COS(CM-YLONGR)
046           COY=C*RAD*(CO2-DATAN(B))-OY
047     C
048           RETURN
049           END
```

```
001     C  SUBROUTINE:  POLUT.LITLR
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE PERPENDICULAR DISTANCE FROM
005     C  THE PUFF CENTER AXIS TO THE RECEPTOR.
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C    XR,YR,ZR = COORDINATES OF RECEPTOR (M)
009     C    XP,YP,ZP = COORDINATES OF PUFF (M)
010     C    XPO,YPO,ZPO = COORDINATES OF PUFF CENTER PROJECTED VERTICALLY
011     C                  DOWN TO GROUND-LEVEL (M)
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C    RL = PERPENDICULAR DISTANCE FROM PUFF CENTER AXIS TO RECEPTOR (M)
015     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
016     C
017           SUBROUTINE LITLR(XR,YR,ZR,XP,YP,ZP,XPO,YPO,ZPO,RL)
018     C
019           DXR=XR-XPO
020           DYR=YR-YPO
021           DZR=ZR-ZPO
022           DXC=XP-XPO
023           DYC=YP-YPO
024           DZC=ZP-ZPO
025           DXR2=DXR**2
026           DYR2=DYR**2
027           DZR2=DZR**2
028           DXC2=DXC**2
029           DYC2=DYC**2
030           R2=DXR2+DYR2+DZR2
031           DPL=DXC2+DYC2
032           T0=DZC/DPL
033           T1=DXC*T0
034           T2=DYC*T0
035           C2=T1**2+T2**2+1.
036           RC=-DXR*T1-DYR*T2+DZR
037           RL=SQRT(R2-(RC**2)/C2)
038     C
039           RETURN
040           END
041
```

```
001     C  SUBROUTINE POLUT.NEWLOC
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE NEW LOCATION OF A
005     C  PUFF ON THE 'ALPHA', 'BETA', 'Z' GRID AFTER TRAVEL TIME 'DELTA T'.
006     C  A TWO STEP ITERATION METHOD IS USED.
007     C
008     C  THE INPUTS ARE:
009     C      DELTAT - TIME STEP (HRS).
010     C      A - ALPHA COORDINATE OF STARTING LOCATION.
011     C      B - BETA COORDINATE OF STARTING LOCATION.
012     C      Z - VERTICAL COORDINATE OF STARTING LOCATION (M AGL).
013     C      CLK - SIMULATION TIME CLOCK (HOUR OF YEAR).
014     C      WOFF - LOGICAL VARIABLE SPECIFYING WHETHER VERTICAL
015     C             VELOCITY IS 'TURNED OFF' AND THE PUFF WILL STAY AT
016     C             A CONSTANT HEIGHT ABOVE GROUND LEVEL. IF .TRUE.
017     C             THEN PUFF WILL FOLLOW TERRAIN WITHOUT USING VERTICAL
018     C             VELOCITY.
019     C
020     C  THE OUTPUTS ARE:
021     C      AN - ALPHA COORDINATE OF NEW LOCATION.
022     C      BN - BETA COORDINATE OF NEW LOCATION.
023     C      ZN - VERTICAL COORDINATE OF NEW LOCATION (M AGL).
024     C      ICODE - CODE FOR EACH PUFF SPECIFYING WHETHER OR NOT PUFF IS
025     C              USEABLE (ICODE=4 MEANS PUFF IS OFF GRID).
026     C      DSM - PUFF INCREMENTAL TRAVEL DISTANCE IN 'DELTA T' (M).
027     C            THIS IS ONLY AN APPROXIMATE MEASURE OF THE TRAVEL
028     C            DISTANCE BECAUSE THE VARIATIONS IN THE TERRAIN HEIGHTS
029     C            ARE NOT FOLLOWED IN BETWEEN THE END POINTS.  A BETTER
030     C            ESTIMATE OF DSM COULD BE OBTAINED BY DIVIDING THE
031     C            SEGMENT IN SMALLER DIVISIONS AND FOLLOWING THE TERRAIN.
032     C
033     C    THIS SUBROUTINE COMPUTES THE HEIGHT AGL OF A PUFF CENTER-OF-MASS.
034     C  IF THE PUFF IS UNIFORMLY MIXED IN THE VERTICAL THEN THE HEIGHT OF THE
035     C  CENTER-OF-MASS IS HALF THE MIXING DEPTH (COMPUTED IN MAIN PROGRAM).
036     C  IF THE PUFF IS NOT UNIFORMLY MIXED IN THE VERTICAL THEN THE HEIGHT
037     C  IS COMPUTED IN ONE OF TWO DIFFERENT WAYS: 1) USING VERTICAL
038     C  VELOCITIES, OR 2) USING THE 'HALF-HEIGHT' CORRECTION.
039     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
040     C
041           SUBROUTINE NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD,ZSH,ELS,
042          +                   A,B,Z,CLK,AN,BN,ZN,ICODE,DSM)
043     C
044           LOGICAL
045          +    WOFF
046     C
047           COMMON /HTADJ/
048          +    CT(6)
049     C
050           FX=3600./(1000.*SLX)
051           FY=3600./(1000.*SLY)
052     C
053     C
054           CALL WINDS (A,B,Z,CLK,U,V,W1,WT1)
055           A1=U*DELTAT*FX+A
056           B1=V*DELTAT*FY+B
057     C              ***IF INTERMEDIATE PUFF OFF GRID SET CODE TO 1000.
058           IF (A1.GT.1..OR.A1.LT.0..OR.B1.GT.1..OR.B1.LT.0.)  THEN
059             ICODE=4
060             RETURN
061           ENDIF
062           CLK=CLK+DELTAT
```

```
063           CALL WINDS (A1,B1,Z,CLK,U,V,W2,WT2)
064           A2=U*DELTAT*FX+A1
065           B2=V*DELTAT*FY+B1
066           AN=.5*(A+A2)
067           BN=.5*(B+B2)
068           IF (AN.GT.1..OR.AN.LT.0..OR.BN.GT.1..OR.BN.LT.0.)  THEN
069             ICODE=4
070             CLK=CLK-DELTAT
071             RETURN
072           ENDIF
073     C
074     C               ***COMPUTE HEIGHT OF PUFF CENTER ABOVE GROUND (M)
075           IF(IPCOD.EQ.2) THEN
076     C               **(HEIGHT IS SET IN MAIN PROGRAM FOR THIS CASE)
077             ZN=Z
078           ELSE
079             IF(WOFF) THEN
080               CALL TETRI(AN,BN,THN,DZA,DZB)
081               HINT=AMIN1(ZSH,THN-ELS)
082               CALL BLPINT(AN,BN,CLK,HMIX,ISTB,USTR,CAPL,WSTR,TSURF,PRESS)
083               ZN=ZSH-(1.-CT(ISTB))*HINT
084             ELSE
085               H1=Z+(W1-WT1)*DELTAT*3600.
086               H2=H1+(W2-WT2)*DELTAT*3600.
087               ZN=(Z+H2)/2.
088             ENDIF
089           ENDIF
090     C
091     C               ***COMPUTE APPROXIMATE DISTANCE TRAVELED.
092           CALL TETRI (A,B,TH,DZA,DZB)
093           HM=Z+TH
094           CALL TETRI (AN,BN,TH,DZA,DZB)
095           HMN=ZN+TH
096           DLTH=(HMN-HM)/1000.
097           DSM=SQRT((SLX*(AN-A))**2.+(SLY*(BN-B))**2.+DLTH**2.)*1000.
098           CLK=CLK-DELTAT
099     C
100           RETURN
101           END
```

```
001     C  SUBROUTINE POLUT.NOCALC
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO DETERMINE WHICH RECEPTORS ARE WITHIN
005     C  A MINIMUM DISTANCE FROM EACH SOURCE.  CONCENTRATIONS FROM THE NEAR
006     C  SOURCE WILL NOT BE CALCULATED AT THESE POINTS.
007     C
008     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
009     C
010           SUBROUTINE NOCALC
011     C
012           PARAMETER
013          +    NSC=5
014     C
015           LOGICAL
016          +    VALL
017     C
018           COMMON /SOURC1/
019          +    EMIS(NSC,2),   NSOURC,        RS(NSC),       TS(NSC),
020          +    VS(NSC),       XSL(NSC),      YSL(NSC),      ZSH(NSC),
021          +    ELS(NSC),      AS(NSC),       BS(NSC),       SIGIN(NSC),
022          +    IDSVAL(NSC),   VALL(NSC),     INDXV(NSC)
023     C
024           COMMON /REC/
025          +    NINDR,         NPTSX(4),      NPTSY(4),      NRGRID,
026          +    RSPAC(4),      STAX(4),       STAY(4),       XIREC(10),
027          +    XREC(25,4),    YIREC(10),     YREC(25,4),    ZIREC(10),
028          +    ZREC(25,25,4)
029     C
030           COMMON /NOC1/
031          +    NGBPT(NSC),    NIBPT(NSC),    IGBAD(100,NSC),IIBAD(10,NSC)
032     C
033     C            ***SPECIFY MINIMUM DISTANCE (M)
034           DMIN=5000.
035     C
036           IF (NRGRID.EQ.0) GO TO 5
037           DO 1 NS=1,NSOURC
038             NGBPT(NS)=0
039         1 CONTINUE
040           DO 4 NG=1,NRGRID
041             IC1=NG*10000
042             DO 3 J=1,NPTSY(NG)
043             IC2=J*100
044             DO 3 I=1,NPTSX(NG)
045             IC=IC1+IC2+I
046             DO 3 NS=1,NSOURC
047             DIST=SQRT((XSL(NS)*1000.-XREC(I,NG))**2+
048          +            (YSL(NS)*1000.-YREC(J,NG))**2)
049             IF(DIST.LE.DMIN) THEN
050               NGBPT(NS)=NGBPT(NS)+1
051               IGBAD(NGBPT(NS),NS)=IC
052             ENDIF
053         3   CONTINUE
054         4 CONTINUE
055     C
056         5 IF (NINDR.EQ.0) RETURN
057           DO 6 NS=1,NSOURC
058             NIBPT(NS)=0
059         6 CONTINUE
060           DO 7 J=1,NINDR
061             DO 7 NS=1,NSOURC
062             DIST=SQRT((XSL(NS)*1000.-XIREC(J))**2+
```

```
063          +              (YSL(NS)*1000.-YIREC(J))**2)
064              IF(DIST.LE.DMIN) THEN
065                NIBPT(NS)=NIBPT(NS)+1
066                IIBAD(NIBPT(NS),NS)=J
067              ENDIF
068            7 CONTINUE
069      C
070              RETURN
071              END
```

```
001     C  SUBROUTINE POLUT.OUT
002     C  DATE APRIL 1985
003     C
004     C   THIS SUBROUTINE IS USED TO OUTPUT A SUMMARY TABLE OF THE
005     C  SPECIFICATIONS OF A RUN POLUT.
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE OUT(IBEG,IEND,IJB,IJE,IHB,IHE,IPRR,NPLT,IDRUN,
010          +               USENAM)
011     C
012           PARAMETER
013          +    NGA=9,          NPO=21,         NPT=50,         NVA=5,
014          +    NSC=5,          NPG=10,         NSG=12
015     C
016           CHARACTER*10
017          +    NAMEG,          NAMER,          NAMES,          IDRUN,
018          +    NASVAL,         USENAM
019     C
020           CHARACTER
021          +    PLTNAM*3,       DISMET(2)*24,   VV*7
022     C
023           LOGICAL
024          +    LUFORM,         WOFF,           VALL
025     C
026           COMMON /MAIN2/
027          +    HA,             HH,             IHH,            IXO,
028          +    IYO,            ROA,            RXY,            SLX,
029          +    SLY
030     C
031           COMMON /TER1/
032          +    ICST,           NXCT,           NYCT,           RGHX(NPT,NPT),
033          +    RGHY(NPT,NPT), THS(NPT,NPT),  XCT,            YCT,
034          +    DXT,            DYT,            HDXT,           HDYT
035     C
036           COMMON /BLPAR1/
037          +  BLP(NPG,NPG,3,7), XCS,            YCS,             HCELLX,
038          +  HCELLY,           NXCS,           NYCS
039     C
040           COMMON /VALLY1/
041          +   COEF(NSG,NVA,3),COXV(NVA),      COYV(NVA),      WSV(NVA),
042          +   SEGX(NSG,NVA),  SEGY(NSG,NVA), NSEG(NVA),      IDVAL(NVA),
043          +   NVALE,          QM(NSG,NSC,2), VSEGL(NSG,NVA),WIDV(NSG,NVA)
044     C
045           COMMON /REC/
046          +    NINDR,          NPTSX(4),       NPTSY(4),       NRGRID,
047          +    RSPAC(4),       STAX(4),        STAY(4),        XIREC(10),
048          +    XREC(25,4),     YIREC(10),      YREC(25,4),     ZIREC(10),
049          +    ZREC(25,25,4)
050     C
051           COMMON /RUNSP/
052          +    LUFORM,         METHOD,         WOFF
053     C
054           COMMON /SOURC1/
055          +    EMIS(NSC,2),    NSOURC,         RS(NSC),        TS(NSC),
056          +    VS(NSC),        XSL(NSC),       YSL(NSC),       ZSH(NSC),
057          +    ELS(NSC),       AS(NSC),        BS(NSC),        SIGIN(NSC),
058          +    IDSVAL(NSC),    VALL(NSC),      INDXV(NSC)
059     C
060           COMMON /NAME1/
061          +    NAMEG(4),       NAMER(10),      NAMES(NSC),     PLTNAM(2),
062          +    NASVAL(NSC)
```

```
063     C
064           DATA DISMET/'          IRWINS GENERAL','MACCREADYS ROUGH TERRAIN'/
065     C
066           LU=6
067     C           ***WRITE OUT TABLE HEADER INFORMATION.
068           WRITE (LU,5)
069           WRITE (LU,49) USENAM
070           WRITE (LU,6) IDRUN
071           WRITE (LU,7) IBEG,IEND
072           WRITE (LU,90)IJB,IHB,IJE,IHE
073     C
074     C           ***WRITE OUT RUN SPECIFICATIONS.
075           WRITE (LU,8)
076           WRITE (LU,9) IPRR
077           IF (LUFORM)  THEN
078           VV='  WAS  '
079           ELSE
080           VV='WAS NOT'
081           ENDIF
082           WRITE (LU,10) VV
083           IF (.NOT.WOFF)  THEN
084           VV='  WAS  '
085           ELSE
086           VV='WAS NOT'
087           ENDIF
088           WRITE (LU,11) VV
089           WRITE (LU,12) DISMET(METHOD)
090     C
091     C           ***WRITE OUT DOMAIN SPECIFICATIONS.
092           WRITE (LU,14)
093           WRITE (LU,15) SLX,SLY
094           X0=REAL(IX0)
095           Y0=REAL(IY0)
096           XE=X0+SLX
097           YE=Y0+SLY
098           WRITE (LU,16) X0,XE,Y0,YE
099     C
100     C           ***WRITE OUT TERRAIN SPECS.
101           WRITE (LU,17)
102           WRITE (LU,19)
103           WRITE (LU,20) ICST,ICST,NXCT,NYCT
104     C
105     C           ***WRITE OUT INPUT MET DATA.
106           WRITE (LU,21)
107           WRITE (LU,22)
108           WRITE (LU,23) SLX,SLY
109           WRITE (LU,24) NXCS,NYCS
110     C
111     C           ***WRITE OUT SOURCE DATA.
112           WRITE (LU,25)
113           NSRP=0
114           NSRA=0
115           DO 50 J=1,NSOURC
116             IF(SIGIN(J).EQ.0.) THEN
117               NSRP=NSRP+1
118             ELSE
119               NSRA=NSRA+1
120             ENDIF
121        50 CONTINUE
122           IF(NSRP.NE.0) THEN
123             WRITE (LU,26) NSRP
124             WRITE (LU,27)
```

```
125             WRITE (LU,28)
126             DO 1 J=1,NSOURC
127               IF(SIGIN(J).EQ.0.) WRITE (LU,29) NAMES(J),XSL(J),YSL(J),
128          +                                     ZSH(J),VS(J),TS(J),RS(J)
129           1   CONTINUE
130             ENDIF
131             IF(NSRA.NE.0) THEN
132               WRITE (LU,56) NSRA
133               WRITE (LU,57)
134               WRITE (LU,58)
135               DO 51 J=1,NSOURC
136                 IF(SIGIN(J).NE.0.) WRITE (LU,59) NAMES(J),XSL(J),YSL(J),
137          +                                       SIGIN(J)
138          51   CONTINUE
139             ENDIF
140      C
141      C                ***WRITE OUT INFORMATION ON SOURCE VALLEYS.
142             WRITE(LU,65)
143             NSRV=0
144             DO 60 J=1,NSOURC
145               IF(VALL(J)) NSRV=NSRV+1
146          60 CONTINUE
147             IF(NSRV.NE.0) THEN
148               WRITE (LU,66) NSRV
149               WRITE (LU,67)
150               WRITE (LU,68)
151               DO 61 J=1,NSOURC
152                 ID=INDXV(J)
153                 IF(VALL(J)) WRITE(LU,69)NAMES(J),NASVAL(J),COXV(ID),COYV(ID)
154          61   CONTINUE
155             ENDIF
156      C
157      C                ***WRITE OUT EMISSIONS DATA.
158             WRITE (LU,30)
159             WRITE (LU,31) NPLT
160             WRITE (LU,32) (PLTNAM(K),K=1,NPLT)
161             IF (NPLT.EQ.2)  THEN
162               WRITE (LU,33)
163             ELSE
164               WRITE (LU,34)
165             ENDIF
166             DO 2 J=1,NSOURC
167               WRITE (LU,35) NAMES(J),(EMIS(J,I),I=1,NPLT)
168           2 CONTINUE
169      C
170      C                ***WRITE OUT RECEPTOR DATA.
171             WRITE (LU,36)
172             IF (NRGRID.NE.0)  THEN
173               WRITE (LU,37) NRGRID
174               WRITE (LU,38)
175               WRITE (LU,39)
176               WRITE (LU,40)
177               DO 3 J=1,NRGRID
178                 STPX=(REAL(NPTSX(J)-1))*RSPAC(J)+STAX(J)
179                 STPY=(REAL(NPTSY(J)-1))*RSPAC(J)+STAY(J)
180                 WRITE (LU,41) J,NAMEG(J),STAX(J),STPX,RSPAC(J),STAY(J),
181          +                    STPY,RSPAC(J)
182           3   CONTINUE
183             ENDIF
184      C
185             IF (NINDR.NE.0)  THEN
186               WRITE (LU,42) NINDR
```

```
187             WRITE (LU,43)
188             WRITE (LU,44)
189             DO 4 J=1,NINDR
190               WRITE (LU,45) J,NAMER(J),XIREC(J)/1000.,YIREC(J)/1000.
191           4   CONTINUE
192             ENDIF
193     C
194             RETURN
195           5 FORMAT (1H1,///,4X,'****SUMMARY RESULTS OF POLUT - POLLUTANT',' CO
196            +NCENTRATION MODEL****')
197          49 FORMAT (1H0,5X,'USER: ',A10)
198           6 FORMAT (6X,'RUN IDENTIFICATION: ',A10)
199           7 FORMAT (6X,'BEGINNING TIME (MMDDYYHH): ',I8.8,'  ENDING TIME: ',
200            +I8.8)
201          90 FORMAT(1H ,T15,'(JULIAN DAY,HOUR): (',I3,',',I2,')',15X,'(',I3
202            +,',',I2,')')
203           8 FORMAT (1H0,/,4X,'**RUN SPECIFICATIONS**')
204           9 FORMAT (1H0,5X,'THE PUFF RELEASE RATE WAS ',I4,' PUFFS PER HOUR.')
205          10 FORMAT (6X,'UNIFORM VERTICAL MIXING ',A7,' ASSUMED.')
206          11 FORMAT(6X,'THE VERTICAL VELOCITY ',A7,' USED TO TRANSPORT PUFFS.')
207          12 FORMAT (6X,'THE METHOD USED TO COMPUTE SIGMA-V AND SIGMA-W WAS:',
208            +/,11X,A24,' FORMULATION.')
209          14 FORMAT (1H0,/,4X,'**DOMAIN**')
210          15 FORMAT (1H0,5X,'REGION SIZE: ',F7.1,'  BY  ',F7.1,'  KILOMETERS')
211          16 FORMAT (6X,'GRAMA COORDINATES (KM):   X= ',F7.1,'  TO  ',F7.1,/,30
212            +X,'  Y= ',F7.1,'  TO  ',F7.1)
213          17 FORMAT (1H0,/,4X,'**INPUT TERRAIN DATA**')
214          19 FORMAT (/,6X,'AVERAGE TERRAIN STATISTICS USED AS INPUT WERE',
215            +' COMPUTED FROM THE',/,6X,'BASE TERRAIN FILE USING THE',
216            + ' FOLLOWING GRID CELL SPECIFICATIONS:')
217          20 FORMAT (11X,'CELL SIZE: ',I4,' BY ',I4,' KILOMETERS',/,11X,'NUMBER
218            + OF CELLS IN X-DIR: ',I4,/,11X,'NUMBER OF CELLS IN Y-DIR: ',I4)
219          21 FORMAT (1H0,/,4X,'**INPUT MET DATA**')
220          22 FORMAT (1H0,5X,'MET DATA SPECIFIED ON FOLLOWING GRID: ')
221          23 FORMAT (1H ,5X,'REGION SIZE: ',F7.1,'  BY  ',F7.1,'  KILOMETERS')
222          24 FORMAT (11X,'NUMBER OF CELLS IN X-DIR: ',I4,/,
223            +         11X,'NUMBER OF CELLS IN Y-DIR: ',I4)
224          25 FORMAT (1H0,/,4X,'**INPUT SOURCE DATA**')
225          26 FORMAT (1H0,5X,'NUMBER OF POINT SOURCES: ',I6)
226          27 FORMAT (6X,'  NAME  ',3X,'X-COR',2X,'Y-COR',2X,'ST HT',
227            +2X,'EX VE',2X,'EX TE',3X,'RAD')
228          28 FORMAT (18X,'(KM)',3X,'(KM)',3X,'(M)',3X,'(M/S)',3X,'(K)',4X,'(M)'
229            +)
230          29 FORMAT (5X,A10,2X,F5.1,2X,F5.1,2X,F5.0,2X,F5.3,2X,F5.0,2X,F5.2)
231          56 FORMAT (1H0,5X,'NUMBER OF AREA SOURCES: ',I6)
232          57 FORMAT (6X,'  NAME  ',3X,'X-COR',2X,'Y-COR',2X,'SIG-Y')
233          58 FORMAT (18X,'(KM)',3X,'(KM)',3X,'(M)')
234          59 FORMAT (5X,A10,2X,F5.1,2X,F5.1,2X,F5.0)
235          65 FORMAT (1H1,///,4X,'**SOURCE VALLEYS**')
236          66 FORMAT (1H0,5X,'NUMBER OF SOURCES LOCATED IN VALLEYS: ',I6)
237          67 FORMAT (6X,'SOURCE NAME',3X,'VALLEY NAME',2X,'X-COR',2X,'Y-COR')
238          68 FORMAT (34X,'(KM)',3X,'(KM)')
239          69 FORMAT (7X,A10,3X,A10,3X,F5.1,2X,F5.1)
240          30 FORMAT (1H0,/,4X,'**INPUT EMISSIONS DATA**')
241          31 FORMAT (1H0,5X,'NUMBER OF POLLUTANTS: ',I6)
242          32 FORMAT (6X,'SOURCE NAME',4X,A3,4X,A3)
243          33 FORMAT (18X,2(2X,'(G/S)'))
244          34 FORMAT (18X,2X,'(G/S)')
245          35 FORMAT (5X,A10,5X,F5.0,2X,F5.0)
246          36 FORMAT (1H0,/,4X,'**RECEPTOR LOCATIONS**')
247          37 FORMAT (1H0,5X,'NUMBER OF RECEPTOR GRIDS: ',I6)
248          38 FORMAT (6X,24X,'X - DIRECTION',10X,'Y - DIRECTION')
```

```
249         39 FORMAT (6X,2X,'GRID',2X,'  NAME  ',5X,'STA X',2X,'STP X',2X,
250           +'SPACE',4X,'STA Y',2X,'STP Y',2X,'SPACE')
251         40 FORMAT (6X,17X,2(5X,'(KM)',3X,'(KM)',3X,'(KM)'))
252         41 FORMAT (6X,3X,I2,2X,A10,2(4X,F5.1,2X,F5.1,2X,F5.1))
253         42 FORMAT (1H0,5X,'NUMBER OF INDIVIDUAL RECEPTORS: ',I6)
254         43 FORMAT (6X,2X,'RECEPTOR',2X,'  NAME  ',4X,'X-CORD',3X,'Y-CORD')
255         44 FORMAT (32X,'(KM)',5X,'(KM)')
256         45 FORMAT (6X,5X,I2,4X,A10,3X,F6.1,3X,F6.1)
257            END
```

```
001     C  SUBROUTINE: POLUT.POLBAS
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE POLYNOMIAL BASIS FUNCTIONS
005     C  'BI' AND 'CI' REFERENCED IN GRAMA REPORT 3 (AUG,1981) TABLE 3.
006     C  THESE ARE USED TO DEFINE THE MASS-CONSISTENT WIND FIELD.
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C  THE INPUTS ARE:
009     C       AL = ALPHA COORDINATE (TRANSFORMED X-COORD.)
010     C       BE = BETA COORDINATE (TRANSFORMED Y-COORD.)
011     C       NTERMS = NUMBER OF TERMS IN EXPANSION
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  THE OUTPUTS ARE:
014     C       BI = ARRAY OF B SUB I TERMS.
015     C       CI = ARRAY OF C SUB I TERMS.
016     C       PBI = ARRAY OF PART. DER. OF BI WITH RESPECT TO ALPHA TERMS.
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  CONSTANT DEFINITIONS:
019     C       E = .01  [MINIMUM VALUE FOR ALPHA AND BETA]
020     C       NDEG = 15  [MAXIMUM DEGREE OF POLYNOMIAL.  THE MAXIMUM
021     C              NUM. OF TERMS IS 136 = (NDEG*(NDEG+1))/2+(NDEG+1)]
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C  PARAMETER DEFINITIONS:
024     C       NPO = NUMBER OF POLYNOMIAL TERMS DESCRIBING WIND COMPONENTS.
025     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
026     C
027           SUBROUTINE POLBAS (AL,BE,NTERMS,BI,CI,PBI)
028     C
029           PARAMETER
030          +    NPO=21
031     C
032           DIMENSION
033          +    BI(NPO),       CI(NPO),       PBI(NPO)
034     C
035     C             ***CONSTANTS
036           E=.01
037           NDEG=15
038     C
039     C              ***COMPUTE BASIS FUNCTIONS FOR ALL TERMS
040           ALPH=AL
041           BETA=BE
042           IF (ALPH.LT.E) ALPH=E
043           IF (BETA.LT.E) BETA=E
044           ICOUNT=0
045           DO 3 IPL=1,NDEG+1
046             DO 2 I=1,IPL
047               ICOUNT=ICOUNT+1
048               N=IPL-1
049               IC=N*(N+1)/2+I
050     C              **U-COMPONENTS
051               NP=N-I+1
052               NQ=I-1
053               BI(IC)=ALPH**NP*BETA**NQ
054     C              **W-COMPONENTS
055               NPP=N-I
056               NQP=I-1
057               NRP=N-I+1
058               PBI(IC)=NRP*ALPH**NPP*BETA**NQP
059     C              **V-COMPONENTS
060               NPB=N-I
061               NQB=I
062               IF (N.LT.1.OR.I.EQ.IPL) GO TO 1
```

```
063               CI(IC)=ALPH**NPB*BETA**NQB
064               AC1=NP
065               AC2=I
066               AC=-AC1/AC2
067               CI(IC)=AC*CI(IC)
068               IF (ICOUNT.EQ.NTERMS) RETURN
069         1     CONTINUE
070         2   CONTINUE
071     C               **LAST TERM IN V-COMPONENTS
072             N=IPL-1
073             IC=N*(N+3)/2+1
074             CI(IC)=ALPH**N
075             AC=-1.
076             CI(IC)=AC*CI(IC)
077             IF (ICOUNT.EQ.NTERMS) RETURN
078         3 CONTINUE
079     C
080           RETURN
081           END
```

```
C  SUBROUTINE POLUT.PRISE
C  DATE: APRIL 1985
C
C    THIS SUBROUTINE COMPUTES FINAL PLUME RISE. IT COMES PRINCIPALLY
C  FROM SUBROUTINE PRISE IN MESOPUFF.
C
C  THE INPUTS ARE:
C      T = AMBIENT AIR TEMPERATURE (DEG K)
C      EV = STACK GAS EXIT VELOCITY (M/SEC)
C      RAD = INSIDE STACK RADIUS (M)
C      TE = STACK GAS EXIT TEMPERATURE (DEG K)
C      UU = AMBIENT WIND SPEED (M/SEC)
C      ISTAB = P-G STABILITY (1,2,3,4,5 OR 6)
C      RLAPSE = POTENTIAL TEMPERATURE LAPSE RATE (DEG K/M)
C      DPBL = MIXING HEIGHT DEPTH (M AGL)
C      DSOURC = HEIGHT OF STACK (M AGL)
C
C  THE OUTPUT IS:
C      RISE = RISE OF PLUME ABOVE STACK HEIGHT (M)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE PRISE (T,EV,RAD,TE,UU,ISTAB,RLAPSE,DPBL,DSOURC,RISE)
C
      S=9.8*RLAPSE/T
      FLUX=9.8*EV*(RAD**2.)*(TE-T)/TE
      IF (FLUX.LE.0.)  THEN
        RISE=0.
        RETURN
      ENDIF
      IF (ISTAB.LE.4) GO TO 2
      IF (UU.GT.1.37) GO TO 1
C             ***FINAL STABLE CALM RISE
      RISE=4.*FLUX**.25/S**.375
      RETURN
    1 CONTINUE
C             ***FINAL STABLE RISE
      RISE=2.6*(FLUX/(S*UU))**.333
      RETURN
C             ***FINAL NEUTRAL, UNSTABLE RISE
    2 XSTAR=AMIN1(34.49*FLUX**.4,14.*FLUX**.625)
      IF (UU.LT.1.37) UU=1.37
      RISE=1.6*FLUX**.333*(3.5*XSTAR)**.667/UU
      ZB=DPBL-DSOURC
      ZB=AMAX1(0.0,ZB)
      IF (RISE.LE.ZB) RETURN
C             ***RISE INTO AN ELEVATED STABLE LAYER
      RSAVE=RISE
      RISE=(1.8*ZB**3+18.75*FLUX/(UU*S))**.333
      RISE=AMIN1(RISE,RSAVE)
      RETURN
C
      END
```

```
001     C  SUBROUTINE: POLUT.ROUINT
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE TERRAIN ROUGHNESS AT A
005     C  POINT.  FIRST A BILINEAR INTERPOLATION TECHNIQUEQUE IS USED TO
006     C  THE ROUGHNESS IN THE X-DIRECTION AND THE ROUGHNESS IN THE
007     C  Y-DIRECTION AT A POINT USING THE GRIDDED ROUGHNESS DATA FROM
008     C  PROGRAM TERRAIN.  THEN THE ROUGHNESS AT A POINT IS DETERMINED
009     C  BY WEIGHTING THE X-DIR ROUGHNESS AND Y-DIR ROUGHNESS ACCORDING
010     C  TO THE DIRECTION OF THE PUFF TRAJECTORY FROM TIME T TO TIME
011     C  T+DELTAT.
012     C    THE INPUTS ARE:
013     C      A0 - ALPHA COORDINATE OF STARTING LOCATION OF TRAJECTORY
014     C      B0 - BETA COORDINATE OF STARTING LOCATION OF TRAJECTORY
015     C      A1 - ALPHA COORDINATE OF ENDING LOCATION OF TRAJECTORY
016     C      B1 - BETA COORDINATE OF ENDING LOCATION OF TRAJECTORY
017     C
018     C    THE OUTPUT IS:
019     C      ROU - MEASURE OF TERRAIN ROUGHNESS (M)
020     C
021     C    THE GRID SPECIFICATION INFORMATION, AND ROUGHNESS MEASURES IN
022     C    THE X-DIR AND Y-DIR ARE IN COMMON BLOCK 'TER1'.
023     C
024     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
025     C
026           SUBROUTINE ROUINT(A0,B0,A1,B1,ROU)
027     C
028           PARAMETER
029          +    NPT=50
030     C
031           COMMON /TER1/
032          +    ICST,          NXCT,          NYCT,          RGHX(NPT,NPT),
033          +    RGHY(NPT,NPT), THS(NPT,NPT),  XCT,           YCT,
034          +    DXT,           DYT,           HDXT,          HDYT
035     C
036           XC=REAL(NXCT)
037           YC=REAL(NYCT)
038     C
039     C              ***COMPUTE X-DIR, Y-DIR WEIGHTING FACTORS AND TRAJECTORY
040     C                 CENTER POINT.
041           HYP=SQRT((A1-A0)**2+(B1-B0)**2)
042           IF(HYP.EQ.0.) THEN
043             FX=.5
044             FY=.5
045             AC=A0
046             BC=B0
047           ELSE
048             THET=ASIN(ABS(B1-B0)/HYP)
049             FY=THET/1.570796
050             FX=1.-FY
051             AC=(A1+A0)/2.
052             BC=(B1+B0)/2.
053           ENDIF
054     C
055     C              ***CALCULATE ARRAY INDICIES.
056           II=INT(XC*AC-0.5)+1
057           JJ=INT(YC*BC-0.5)+1
058     C              ***CALC WEIGHTING FACTORS FOR BILINEAR INTERPOLATION.
059           X2=AC*XC-REAL(II)+0.5
060           X1=1.-X2
061           Y2=BC*YC-REAL(JJ)+0.5
062           Y1=1.-Y2
```

```
063     C              ***CHECK IF POINT IS ON HALF-CELL BORDERS OF DOMAIN
064           CELLX=1.0/XC
065           CELLY=1.0/YC
066           HCELLX=0.5*CELLX
067           HCELLY=0.5*CELLY
068           IF (AC.LT.HCELLX) GO TO 1
069           IF (BC.LT.HCELLY) GO TO 4
070           IF (AC.GE.1.-HCELLX) GO TO 5
071           IF (BC.GE.1.-HCELLY) GO TO 8
072     C              ***COMPUTE TERRAIN ROUGHNESS AT ALPHA AND BETA.
073           RX=X1*Y1*RGHX(II,JJ)+X2*Y1*RGHX(II+1,JJ)+X1*Y2*RGHX(II,JJ+1)+
074          +X2*Y2*RGHX(II+1,JJ+1)
075           RY=X1*Y1*RGHY(II,JJ)+X2*Y1*RGHY(II+1,JJ)+X1*Y2*RGHY(II,JJ+1)+
076          +X2*Y2*RGHY(II+1,JJ+1)
077           GO TO 9
078         1 II=1
079           IF (BC.LT.HCELLY) GO TO 2
080           IF (BC.GE.1.-HCELLY) GO TO 3
081           RX=Y2*RGHX(II,JJ+1)+Y1*RGHX(II,JJ)
082           RY=Y2*RGHY(II,JJ+1)+Y1*RGHY(II,JJ)
083           GO TO 9
084         2 JJ=1
085           RX=RGHX(II,JJ)
086           RY=RGHY(II,JJ)
087           GO TO 9
088         3 JJ=NYCT
089           RX=RGHX(II,JJ)
090           RY=RGHY(II,JJ)
091           GO TO 9
092         4 JJ=1
093           RX=X2*RGHX(II+1,JJ)+X1*RGHX(II,JJ)
094           RY=X2*RGHY(II+1,JJ)+X1*RGHY(II,JJ)
095           GO TO 9
096         5 II=NXCT
097           IF (BC.LT.HCELLY) GO TO 6
098           IF (BC.GE.1.0-HCELLY) GO TO 7
099           RX=Y2*RGHX(II,JJ+1)+Y1*RGHX(II,JJ)
100           RY=Y2*RGHY(II,JJ+1)+Y1*RGHY(II,JJ)
101           GO TO 9
102         6 JJ=1
103           RX=RGHX(II,JJ)
104           RY=RGHY(II,JJ)
105           GO TO 9
106         7 JJ=NYCT
107           RX=RGHX(II,JJ)
108           RY=RGHY(II,JJ)
109           GO TO 9
110         8 JJ=NYCT
111           RX=X2*RGHX(II+1,JJ)+X1*RGHX(II,JJ)
112           RY=X2*RGHY(II+1,JJ)+X1*RGHY(II,JJ)
113         9 CONTINUE
114     C
115           ROU=FX*RX+FY*RY
116     C
117           RETURN
118           END
```

```
001     C  SUBROUTINE: POLUT.SHSLP
002     C  DATE: APRIL 1985
003     C
004     C     SUBROUTINE TO EVALUATE THE PARTIAL DERIVITIVE OF 'SMALL H'
005     C  WITH RESPECT TO ALPHA (DHA) AND WITH RESPECT TO BETA (DHB).
006     C  THE RESULTS ARE USED IN THE COMPUTATION OF THE MASS-CONSISTENT
007     C  WINDS.  'SMALL H' IS THE DISTANCE BETWEEN THE TERRAIN SURFACE
008     C  AND THE UPPER BOUNDARY TO THE FLOW.
009     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
010     C  THE INPUTS ARE:
011     C       IHH = OPTION NUMBER IDENTIFYING FUNCTIONAL FORM OF 'SMALL H'.
012     C             (REFER TO SUBROUTINE SMALLH FOR EXPLAINATION)
013     C       DZA = TERRAIN SLOPE WITH RESPECT TO ALPHA.
014     C       DZB = TERRAIN SLOPE WITH RESPECT TO BETA.
015     C       A = ALPHA (TRANSFORMED X-COORDINATE).
016     C       B = BETA (TRANSFORMED Y-COORDINATE).
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       DHA = SLOPE OF SMALL H WITH RESPECT TO ALPHA.
020     C       DHB = SLOPE OF SMALL H WITH RESPECT TO BETA.
021     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
022     C
023           SUBROUTINE SHSLP (IHH,DZA,DZB,A,B,DHA,DHB)
024     C
025           GO TO (1,2,3,4), IHH
026     C
027     C                **SMALL H = CONSTANT.
028         1 DHA=0.
029           DHB=0.
030           GO TO 5
031     C
032     C                **BIG H = CONSTANT.
033         2 DHA=-DZA
034           DHB=-DZB
035           GO TO 5
036     C
037     C                **SMALL H SOME FUNCTIONAL FORM OF A AND B.
038     C                **ANALYTICAL FORM ON P89 OF GRAMA 3 IN NOW.
039         3 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 1
040           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 1
041           DHA=100.*EXP(A-B)
042           DHB=-100.*EXP(A-B)
043           GO TO 5
044     C
045     C                **BIG H IS SOME FUNCTIONAL FORM OF A AND B.
046     C                **DHA=BIGH,A(A,B) - DZA
047     C                **DHB=BIGH,B(A,B) - DZB
048         4 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 2
049           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 2
050           GO TO 5
051     C
052         5 RETURN
053           END
```

```
C  SUBROUTINE POLUT.SIGGS
C  APRIL 1985
C
C    THIS SUBROUTINE IS USED TO COMPUTE SIGMAY AND SIGMAZ.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SIGGS(N,NG,NS,CLK,DELTAT,ID)
C
      PARAMETER
     +    NPU=100,        NSC=5,          NSG=12,         NVA=5
C
      LOGICAL
     +    LUFORM,         WOFF
C
      COMMON /PUFF1/
     +    A(NPU,NSG,NSC),      B(NPU,NSG,NSC),       IPCOD(NPU,NSG,NSC),
     +    SGPRY(NPU,NSC),      SGDPY(NPU,NSG,NSC),  SGDGY(NPU,NSG,NSC),
     +    SGSHY(NPU,NSG,NSC),  SGTOY(NPU,NSG,NSC),  SGPRZ(NPU,NSC),
     +    SGDZ(NPU,NSG,NSC),   SGTOZ(NPU,NSG,NSC),  STOTM(NPU,NSG,NSC),
     +    TIME(NPU,NSG,NSC),   Z(NPU,NSG,NSC),      ZMIX(NPU,NSG,NSC),
     +    QTP(NPU,NSG,NSC,2),  IQCOD(NPU,NSG,NSC),  NPUFFS(NSC),
     +    NPUFFV(NSC)
C
      COMMON /TPDT/
     +    ANQ(2),         BNQ(2),         DSMQ(2),        HTMIXQ(2),
     +    IPCODQ(2),      SYDPQ(2),       SYDGQ(2),       SYSHQ(2),
     +    SYTOQ(2),       SZDQ(2),        SZTOQ(2),       ZNQ(2),
     +    ANQS(2),        BNQS(2),        DSMQS(2),       ZNQS(2),
     +    AF(4),          BF(4),          QF(2)
C
      COMMON /RUNSP/
     +    LUFORM,         METHOD,         WOFF
C
      ED=15000.
C
C              ***SPECIFY MIDPOINT OF PUFF TRAJECTORY OVER DELTA T.
      AI=(A(N,NG,NS)+ANQ(ID))/2.
      BI=(B(N,NG,NS)+BNQ(ID))/2.
      CLKI=CLK+DELTAT/2.
C
      CALL BLPINT (AI,BI,CLKI,RMIX,ISTB,USTR,CAPL,WSTR,TSURF,PRESS)
      HTMIXQ(ID)=AMAX1(ZMIX(N,NG,NS),RMIX)
      IPCODQ(ID)=IPCOD(N,NG,NS)
      IF (IPCODQ(ID).EQ.3.AND.ZNQ(ID).LE.HTMIXQ(ID)) IPCODQ(ID)=1
      IF (LUFORM.AND.IPCODQ(ID).EQ.1) IPCODQ(ID)=2
      IF (IPCODQ(ID).EQ.2) ZNQ(ID)=HTMIXQ(ID)/2.
      ZI=(Z(N,NG,NS)+ZNQ(ID))/2.
      CALL WINDS (AI,BI,ZI,CLKI,U,V,W,WT)
      SPEED=SQRT(U**2+V**2)
      SPRZ=SGPRZ(N,NS)
      SPRY=SGPRY(N,NS)
      IF(NG.GT.1) THEN
        SPRZ=0.
        SPRY=0.
      ENDIF
      IF (IPCODQ(ID).EQ.2) GO TO 30
      CALL SIGMAZ (METHOD,USTR,WSTR,HTMIXQ(ID),CAPL,ISTB,A(N,NG,NS),
     +             B(N,NG,NS),ANQ(ID),BNQ(ID),ZI,SPEED,TIME(N,NG,NS),
     +             SPRZ,SGDZ(N,NG,NS),DELTAT,SZDQ(ID),SZTOQ(ID))
      IF (IPCODQ(ID).EQ.1.AND.SZTOQ(ID).GE.2.*HTMIXQ(ID)) IPCODQ(ID)=2
   30 CALL SIGMAY (METHOD,USTR,HTMIXQ(ID),CAPL,ISTB,A(N,NG,NS),
```

```
063           +              B(N,NG,NS),ANQ(ID),BNQ(ID),DSMQ(ID),ZI,SPEED,
064           +              TIME(N,NG,NS),SPRY,SGDPY(N,NG,NS),SGDGY(N,NG,NS),
065           +              SGSHY(N,NG,NS),SZTOQ(ID),DELTAT,IPCODQ(ID),CLKI,ED,
066           +              SYDPQ(ID),SYDGQ(ID),SYSHQ(ID),SYTOQ(ID))
067     C
068            RETURN
069            END
```

```
001     C  SUBROUTINE POLUT.SIGMAV
002     C  DATE APRIL 1985
003     C
004     C    SUBROUTINE TO DETERMINE THE STANDARD DEVIATION OF THE
005     C   HORIZONTAL WIND GIVEN:
006     C      METH = METHOD OF COMPUTING SIGMA V.
007     C           = 1 USE IRWIN SCHEME
008     C           = 2 USE MACCREADY'S SCHEME FOR IS=4,5,6 AND IRWIN'S
009     C               SCHEME FOR IS=1,2,3
010     C      USTAR = SURFACE FRICTION VELOCITY (M/S)
011     C      HMIX = HEIGHT OF MIXING LAYER (M AGL)
012     C      OBUKL = MONIN-OBUKOV LENGTH (M)
013     C      IS = P-G-T STABILITY
014     C      A0 = ALPHA COORDINATE OF STARTING LOCATION OF TRAJECTORY
015     C      B0 = BETA COORDINATE OF STARTING LOCATION OF TRAJECTORY
016     C      A1 = ALPHA COORDINATE OF ENDING LOCATION OF TRAJECTORY
017     C      B1 = BETA COORDINATE OF ENDING LOCATION OF TRAJECTORY
018     C      SP = WIND SPEED (M/S)
019     C      HAG = HEIGHT OF DESIRED POINT (M AGL)
020     C    THE OUTPUT IS:
021     C      SIGV = STANDARD DEVIATION OF HORIZONTAL COMPONENT OF
022     C             WINDS (M/S)
023     C
024     C    SUBROUTINE ROUINT IS CALLED WHICH SUPPLIES TERRAIN ROUGHNESS TO
025     C  SIGV CALCULATION.
026     C
027     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
028     C
029           SUBROUTINE SIGMAV(METH,USTAR,HMIX,OBUKV,IS,A0,B0,A1,B1,
030          +                  SP,HAG,SIGV)
031     C
032           IF(OBUKV.LT.0.)THEN
033           SIGV=USTAR*((12.-0.5*HMIX/OBUKV)**0.333)
034           ELSE
035             IF(METH.EQ.1)THEN
036             SIGV=1.78*USTAR
037             ELSE
038             CALL ROUINT(A0,B0,A1,B1,ROU)
039               IF(IS.EQ.6)THEN
040               SIGV=0.186*(ROU**0.35)*(SP**0.18)*(1./HAG**0.13)
041               ELSE
042               SIGV=0.231*(ROU**0.28)*(SP**0.06)*(1./HAG**0.04)
043               ENDIF
044             ENDIF
045           ENDIF
046           IF(SIGV.LE.0.) SIGV=.01
047           RETURN
048           END
```

```
001     C  SUBROUTINE POLUT.SIGMAW
002     C  DATE APRIL 1985
003     C
004     C    SUBROUTINE TO DETERMINE THE STANDARD DEVIATION OF THE
005     C   VERTICAL WIND GIVEN:
006     C      METH = METHOD OF COMPUTING SIGMA V.
007     C           = 1 USE IRWIN SCHEME
008     C           = 2 USE MACCREADY'S SCHEME FOR IS=4,5,6 AND IRWIN'S
009     C             SCHEME FOR IS=1,2,3
010     C      USTAR = SURFACE FRICTION VELOCITY (M/S)
011     C      WSTAR = CONVECTIVE VELOCITY (M/S)
012     C      HMIX = HEIGHT OF MIXING LAYER (M AGL)
013     C      OBUKV = MONIN-OBUKOV LENGTH (M)
014     C      IS = P-G-T STABILITY
015     C      A0 = ALPHA COORDINATE OF STARTING LOCATION OF TRAJECTORY
016     C      B0 = BETA COORDINATE OF STARTING LOCATION OF TRAJECTORY
017     C      A1 = ALPHA COORDINATE OF ENDING LOCATION OF TRAJECTORY
018     C      B1 = BETA COORDINATE OF ENDING LOCATION OF TRAJECTORY
019     C      SPEED = WIND SPEED (M/S)
020     C      HAG = HEIGHT OF DESIRED POINT (M AGL)
021     C    THE OUTPUT IS:
022     C      SIGW = STANDARD DEVIATION OF VERTICAL COMPONENT OF
023     C             WINDS (M/S)
024     C
025     C    SUBROUTINE ROUINT IS CALLED WHICH SUPPLIES TERRAIN ROUGHNESS TO
026     C  SIGW CALCULATION.
027     C
028     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
029     C
030           SUBROUTINE SIGMAW(METH,USTAR,WSTAR,HMIX,OBUKV,IS,A0,B0,A1,B1,
031          +                  SPEED,HAG,SIGW)
032     C
033           IF(OBUKV.LE.0.)THEN
034             ZOH=HAG/HMIX
035             IF(ZOH.LE..03) THEN
036               SIGW=WSTAR*1.342*ZOH**.33
037             ELSE
038               IF(ZOH.LE..4) THEN
039                 SIGW=WSTAR*.763*ZOH**.175
040               ELSE
041                 IF(ZOH.LE..96) THEN
042                   SIGW=WSTAR*.722*(1.-ZOH)**.207
043                 ELSE
044                   SIGW=WSTAR*.37
045                 ENDIF
046               ENDIF
047             ENDIF
048           ELSE
049             IF(METH.EQ.1)THEN
050               ZOL=HAG/OBUKV
051               PHIM=1.+5.*ZOL
052               IF(ZOL.LE.2.) THEN
053                 FM=.4*(1.+3.39*ZOL-.25*ZOL**2)
054               ELSE
055                 FM=.4*(6.78+2.39*(ZOL-2.))
056               ENDIF
057               SIGW=(((PHIM-ZOL)/(1.2*FM))**.333)*USTAR
058             ELSE
059             CALL ROUINT(A0,B0,A1,B1,ROU)
060               IF(IS.EQ.6)THEN
061               SIGW=0.123*(ROU**0.42)*(SPEED**0.22)*(1./HAG**0.17)
062               ELSE
```

```
063               SIGW=0.251*(ROU**0.26)*(SPEED**0.05)*(1./HAG**0.03)
064               ENDIF
065             ENDIF
066           ENDIF
067           IF(SIGW.LE.0.) SIGW=.01
068           RETURN
069           END
```

```
001     C  SUBROUTINE: POLUT.SIGMAY
002     C  DATE:  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE HORIZONTAL SPREAD
005     C  DUE TO ATMOSPHERIC TURBULENCE.  THE SIGMA Y IS THE GREATER OF
006     C  PLUME SPREAD COMPUTED USING PASQUILL'S GENERAL FORM, OR PUFF
007     C  SPREAD COMPUTED USING GIFFORD'S EQUATIONS.
008     C    THE INPUTS ARE:
009     C      1) FOR THE COMPUTATION OF SIGMA V:
010     C         METH = METHOD FOR COMPUTING SIGMA V.
011     C              = 1 USE IRWIN SCHEME
012     C              = 2 USE MACCREADY'S SCHEME FOR IS=4,5,6 AND IRWIN'S
013     C                  SCHEME FOR IS=1,2,3
014     C         USTAR = SURFACE FRICTION VELOCITY (M/S)
015     C         HMIX= HEIGHT OF MIXING LAYER (M AGL)
016     C         OBUKV = MONIN-OBUKOV LENGTH (M)
017     C         IS = P-G-T STABILITY (1,2,3,4,5 OR 6)
018     C         A0 = ALPHA COORDINATE OF STARTING LOCATION OF TRAJECTORY
019     C         B0 = BETA COORDINATE OF STARTING LOCATION OF TRAJECTORY
020     C         A1 = ALPHA COORDINATE OF ENDING LOCATION OF TRAJECTORY
021     C         B1 = BETA COORDINATE OF ENDING LOCATION OF TRAJECTORY
022     C         SP = WIND SPEED (M/S)
023     C         HAG = HEIGHT OF DESIRED POINT (M AGL)
024     C      2) FOR COMPUTATION OF VIRTUAL TIMES:
025     C         TIMO = INITIAL GUESS AT VIRTUAL TIME, LET IT BE THE
026     C                ELAPSED SIMULATION TIME T  (S)
027     C         BYOP = SIGMA Y DUE TO ATMOS. TURB. FOR PASQUILL'S
028     C                GENERAL FORM, AT TIME T (M)
029     C         BYOG = SIGMA Y DUE TO ATMOS. TURB. FOR GIFFORD'S
030     C                PUFF FORMULATION, AT TIME T (M)
031     C      3) OTHER INPUTS:
032     C         DELTAT = SIMULATION TIME STEP (S)
033     C         ED = LARGE-SCALE EDDY DIFFUSIVITY (M**2/S)
034     C    THE OUTPUTS ARE:
035     C      BYP = SIGMA Y AT TIME T+DELTAT USING PASQUILL'S FORM (M)
036     C      BYG = SIGMA Y AT TIME T+DELTAT USING GIFFORD'S FORM (M)
037     C      BY = SIGMA Y DUE TO ATMOS. TURB. AT TIME T+DELTAT, (M)
038     C
039     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
040     C
041           SUBROUTINE SIGMAY(METH,USTAR,HMIX,OBUKV,IS,A0,B0,A1,B1,DX,HAG,
042          +                  SPD,TIMO,AYPRO,BYOP,BYOG,CYSHO,SZD,DELTAT,
043          +                  IPCOD,CLK,ED,BYP,BYG,CYSH,SIGY)
044     C
045           DOUBLE PRECISION
046          +    SED,          SSIGV,        AA
047     C
048           COMMON /MAIN2/
049          +    HA,           HH,           IHH,          IXO,
050          +    IYO,          ROA,          RXY,          SLX,
051          +    SLY
052     C
053     C           ***DETERMINE STD. DEV. OF HORIZONTAL WINDS (SIGMA V)
054           CALL SIGMAV(METH,USTAR,HMIX,OBUKV,IS,A0,B0,A1,B1,
055          +            SPD,HAG,SIGV)
056     C
057     C           ***COMPUTE PLUME SIGMA Y USING PASQUILL'S GENERAL FORM
058     C           ***COMPUTE VIRTUAL TIME
059           TIMM=TIMO*3600.
060           CALL YVIRT(TIMM,SIGV,BYOP,VT)
061           TIM=VT+DELTAT*3600.
062     C           ***COMPUTE UNIVERSAL FUNCTION USING DRAXLER'S EQNS.
```

```
063           FUNC=1./(1.+.9*SQRT(TIM/1000.))
064     C
065           BYP=SIGV*TIM*FUNC
066     C
067     C        ***COMPUTE PUFF SIGMA Y USING GIFFORD'S EQUATION.
068     C        ***COMPUTE VIRTUAL TIME.
069           TIMM=TIMO*3600.
070           CALL YVIRTG(TIMM,SIGV,ED,BYOG,VT)
071           TIM=VT+DELTAT*3600.
072     C
073           SED=ED**2
074           SSIGV=SIGV**2
075           AA=(1.-1./DEXP(SSIGV/ED*TIM))
076           AARG=2.*ED*TIM-2.*AA*SED/SSIGV-AA**2*SED/SSIGV
077           IF(AARG.LE.0) THEN
078             BYG=BYOG
079           ELSE
080             BYG=SQRT(AARG)
081           ENDIF
082     C
083     C          ***DETERMINE LARGER OF BYPLUM AND BYPUF.
084           BY=MAX(BYP,BYG)
085     C
086     C                ***COMPUTE SHEAR SIGMA
087           IF (IPCOD.EQ.2)  THEN
088             ZL=1.
089             ZH=HMIX
090           ELSE
091             ZL=HAG-SZD
092             IF (ZL.LT.0.) ZL=1.
093             ZH=HAG+SZD
094           ENDIF
095           A=(A0+A1)/2.
096           B=(B0+B1)/2.
097           CALL TETRI (A,B,THH,DZA,DZB)
098           CALL SMALLH (IHH,HH,THH,A,B,SH)
099           IF (ZL.GT.SH) ZL=SH
100           IF (ZH.GT.SH) ZH=SH
101           CALL WINDS (A,B,ZL,CLK,UL,VL,W,WT)
102           CALL WINDS (A,B,ZH,CLK,UH,VH,W,WT)
103           ADB=UL*UH+VL*VH
104           AM=SQRT(UL**2+VL**2)
105           BM=SQRT(UH**2+VH**2)
106           ARG=ADB/(AM*BM)
107           IF(ARG.NE.1.) THEN
108             THET=ACOS(ARG)
109           ELSE
110             THET=0.
111           ENDIF
112           CYSH=.173*THET*DX+CYSHO
113     C
114     C               ***COMPUTE COMPLETE SIGMA Y.
115           SIGY=SQRT(AYPRO**2+BY**2+CYSH**2)
116           RETURN
117           END
118
```

```
001     C  SUBROUTINE: POLUT.SIGMAZ
002     C  DATE:  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE VERTICAL SPREAD
005     C  DUE TO ATMOSPHERIC TURBULENCE.
006     C    THE INPUTS ARE:
007     C      1) FOR THE COMPUTATION OF SIGMA W:
008     C         METH = METHOD FOR COMPUTING SIGMA V.
009     C              = 1 USE IRWIN SCHEME
010     C              = 2 USE MACCREADY'S SCHEME FOR IS=4,5,6 AND IRWIN'S
011     C                  SCHEME FOR IS=1,2,3
012     C         USTAR = SURFACE FRICTION VELOCITY (M/S)
013     C         WSTAR = CONVECTIVE VELOCITY (M/S)
014     C         HMIX= HEIGHT OF MIXING LAYER (M AGL)
015     C         OBUKV = MONIN-OBUKOV LENGTH (M)
016     C         IS = P-G-T STABILITY (1,2,3,4,5 OR 6)
017     C         A0 = ALPHA COORDINATE OF STARTING LOCATION OF TRAJECTORY
018     C         B0 = BETA COORDINATE OF STARTING LOCATION OF TRAJECTORY
019     C         A1 = ALPHA COORDINATE OF ENDING LOCATION OF TRAJECTORY
020     C         B1 = BETA COORDINATE OF ENDING LOCATION OF TRAJECTORY
021     C         SP = WIND SPEED (M/S)
022     C         HAG = HEIGHT OF DESIRED POINT (M AGL)
023     C       2) FOR COMPUTATION OF VIRTUAL TIMES:
024     C         TIMO = INITIAL GUESS AT VIRTUAL TIME, LET IT BE THE
025     C                ELAPSED SIMULATION TIME T  (S)
026     C         BZO = SIGMA Z DUE TO ATMOS. TURB. AT TIME T (M)
027     C       3) OTHER INPUTS:
028     C         DELTAT = SIMULATION TIME STEP (S)
029     C     THE OUTPUTS ARE:
030     C       BZ = SIGMA Z AT TIME T+DELTAT USING PASQUILL'S FORM (M)
031     C       SIGZ = TOTAL SIGMA Z AT TIME T+DELTAT, (M)
032     C
033     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
034     C
035           SUBROUTINE SIGMAZ(METH,USTAR,WSTAR,HMIX,OBUKV,IS,A0,B0,A1,B1,
036          +                  HAG,SPD,TIMO,AZPRO,BZO,DELTAT,BZ,SIGZ)
037     C
038           DIMENSION
039          +    RKZ(6)
040     C
041           DATA RKZ/50.,30.,15.,7.,3.,1./
042     C
043           TIMM=TIMO*3600.
044           DLT=DELTAT*3600.
045     C
046     C          ***DETERMINE STD. DEV. OF VERTICAL WINDS (SIGMA W)
047           CALL SIGMAW(METH,USTAR,WSTAR,HMIX,OBUKV,IS,A0,B0,A1,B1,
048          +            SPD,HAG,SIGW)
049     C
050     C          ***COMPUTE PLUME SIGMA Z USING HEFTER'S FORM
051           IF(TIMM.GT.20000.) THEN
052             BZ=BZO+SQRT(2.*RKZ(IS))*DLT/(2.*SQRT(TIMM))
053           ELSE
054     C
055     C          ***COMPUTE SIGMAZ USING PASQUILL'S GENERAL FORM
056     C          **COMPUTE VIRTUAL TIME
057             CALL ZVIRT(TIMM,IS,SIGW,BZO,VT)
058             TIM=VT+DLT
059     C          ***COMPUTE UNIVERSAL FUNCTION USING DRAXLER'S EQNS.
060             IF(IS.LE.4) THEN
061               C3=500.
062             ELSE
```

```
063               C3=50.
064             ENDIF
065             FZ=1./(1.+.9*SQRT(TIM/C3))
066             BZ=SIGW*TIM*FZ
067           ENDIF
068     C
069     C                ***COMPUTE COMPLETE SIGMA Z.
070           SIGZ=SQRT(AZPRO**2+BZ**2)
071     C
072           RETURN
073           END
074
```

```
001     C  PROGRAM: MET.SMALLH
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO DETERMINE 'SMALL H' WHERE 'SMALL H'
005     C  IS THE DISTANCE BETWEEN THE TERRAIN SURFACE AND THE UPPER
006     C  BOUNDARY TO THE FLOW.  'SMALL H' = 'BIG H' - 'TERRAIN HGHT',
007     C  WHERE 'BIG H' IS THE HEIGHT (M MSL) OF THE UPPER BOUNDARY.
008     C  'SMALL H' IS USED IN THE COMPUTATION OF THE MASS-CONSISTENT
009     C  WINDS.
010     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
011     C  THE INPUTS ARE:
012     C       IHH = 1  'SMALL H' = CONSTANT
013     C           = 2  'BIG H' = CONSTANT
014     C           = 3  'SMALL H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
015     C           = 4  'BIG H' = SOME FUNCTIONAL FORM OF ALPHA AND BETA
016     C       HH = VALUE FOR 'SMALL H' OR 'BIG H' DEPENDING ON WHETHER
017     C            IHH=1 OR IHH=2, RESPECTIVELY. (M) [IF STATION OUTSIDE
018     C            DOMAIN FOR IHH=3 OR 4, THEN 'SMALL H' OR 'BIG H' IS
019     C            SET EQUAL TO HH.]
020     C       TH = TERRAIN HEIGHT (M MSL)
021     C       A = ALPHA (TRANSFORMED X-COORDINATE)
022     C       B = BETA (TRANSFORMED Y-COORDINATE)
023     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
024     C  THE OUTPUTS ARE:
025     C       SH = 'SMALL H' (M)
026     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
027     C
028           SUBROUTINE SMALLH (IHH,HH,TH,A,B,SH)
029     C
030           GO TO (1,2,3,4), IHH
031     C
032     C                **SMALL H = CONSTANT.
033         1 SH=HH
034           GO TO 5
035     C
036     C                **BIG H = CONSTANT.
037         2 SH=HH-TH
038           GO TO 5
039     C
040     C                **FUNCTIONAL FORM OF SMALL H.
041     C                **THE ANALYTICAL FUNCTION ON P89 OF GRAMA 3 IS IN NOW.
042         3 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 1
043           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 1
044           SH=2000.+100.*EXP(A-B)
045           GO TO 5
046     C
047     C                **FUNCTIONAL FORM OF BIG H.
048     C                **MUST INSERT H(A,B) WHERE SH=H(A,B)-TH.
049         4 IF (A.LT.0.0.OR.A.GT.1.0) GO TO 2
050           IF (B.LT.0.0.OR.B.GT.1.0) GO TO 2
051           GO TO 5
052     C
053         5 RETURN
054           END
```

```
001     C  SUBROUTINE: POLUT.TETRI
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO INTERPOLATE THE TERRAIN HEIGHT BETWEEN
005     C  GRID POINTS.  A SIXTEEN POINT INTERPOLATION TECHNIQUE IS USED.
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C    THE INPUTS ARE:
008     C      XG - ALPHA COORDINATE OF DESIRED POINT.
009     C      YG - BETA COORDINATE OF DESIRED POINT.
010     C
011     C    THE OUTPUTS ARE:
012     C      TH - TERRAIN HEIGHT AT POINT (M MSL).
013     C      DZA - SLOPE IN ALPHA (X) DIRECTION ( M/[DELTA ALPHA] ).
014     C      DZB - SLOPE IN BETA (Y) DIRECTION ( M/[DELTA BETA] ).
015     C
016     C    THE GRID SPECIFICATION INFORMATION IS IN COMMON BLOCK 'MAIN2'.
017     C    THE ARRAY OF TERRAIN HEIGHTS ARE IN COMMON BLOCK 'TETRI1'.
018     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
019     C
020           SUBROUTINE TETRI (XG,YG,TH,DZA,DZB)
021     C
022           PARAMETER
023          +    NPT=50
024     C
025           COMMON /MAIN2/
026          +    HA,             HH,             IHH,            IX0,
027          +    IY0,            ROA,            RXY,            SLX,
028          +    SLY
029     C
030           COMMON /TER1/
031          +    ICST,           NXCT,           NYCT,           RGHX(NPT,NPT),
032          +    RGHY(NPT,NPT), THS(NPT,NPT),    XCT,            YCT,
033          +    DXT,            DYT,            HDXT,           HDYT
034     C
035     C
036     C              ***CALCULATE ARRAY INDICIES.
037           II=INT(XCT*XG-0.5)+1
038           JJ=INT(YCT*YG-0.5)+1
039     C
040     C              ***COMPUTE TERRAIN HEIGHT AT POINT.
041           AI=REAL(II-1)*DXT+HDXT
042           BI=REAL(JJ-1)*DYT+HDYT
043           AC=(XG-AI)/DXT
044           BC=(YG-BI)/DYT
045     C
046           F1=1.-BC
047           F2=2.-BC
048           F3=1.+BC
049           B1=-0.166667*F1*F2*BC
050           B2=0.5*F1*F2*F3
051           B3=0.5*F2*F3*BC
052           B4=-0.166667*F1*F3*BC
053     C
054           F1=1.-AC
055           F2=2.-AC
056           F3=1.+AC
057           A1=-0.166667*F1*F2*AC
058           A2=0.5*F1*F2*F3
059           A3=0.5*F2*F3*AC
060           A4=-0.166667*F1*F3*AC
061     C
062           IM1=II-1
```

```
063           IF(IM1.LT.1) IM1=1
064           IP1=II+1
065           IF(IP1.GT.NXCT) IP1=NXCT
066           IP2=II+2
067           IF(IP2.GT.NXCT) IP2=NXCT
068           JM1=JJ-1
069           IF(JM1.LT.1) JM1=1
070           JP1=JJ+1
071           IF(JP1.GT.NYCT) JP1=NYCT
072           JP2=JJ+2
073           IF(JP2.GT.NYCT) JP2=NYCT
074     C
075           TH1=B1*(A1*THS(IM1,JM1)+A2*THS(II,JM1)+A3*THS(IP1,JM1)+
076          +        A4*THS(IP2,JM1))
077           TH2=B2*(A1*THS(IM1,JJ)+A2*THS(II,JJ)+A3*THS(IP1,JJ)+
078          +        A4*THS(IP2,JJ))
079           TH3=B3*(A1*THS(IM1,JP1)+A2*THS(II,JP1)+A3*THS(IP1,JP1)+
080          +        A4*THS(IP2,JP1))
081           TH4=B4*(A1*THS(IM1,JP2)+A2*THS(II,JP2)+A3*THS(IP1,JP2)+
082          +        A4*THS(IP2,JP2))
083           TH=TH1+TH2+TH3+TH4
084     C
085     C               **COMPUTE SLOPE WITH RESPECT TO ALPHA
086           IF(II.EQ.1) THEN
087             DZA=(THS(2,JJ)-THS(1,JJ))/DXT
088           ELSE
089             IF(II.EQ.NXCT) THEN
090               DZA=(THS(NXCT,JJ)-THS(NXCT-1,JJ))/DXT
091             ELSE
092               DZA=(THS(II+1,JJ)-THS(II-1,JJ))/(2.*DXT)
093             ENDIF
094           ENDIF
095     C
096     C               **COMPUTE SLOPE WITH RESPECT TO BETA
097           IF(JJ.EQ.1) THEN
098             DZB=(THS(II,2)-THS(II,1))/DYT
099           ELSE
100             IF(JJ.EQ.NYCT) THEN
101               DZB=(THS(II,NYCT)-THS(II,NYCT-1))/DYT
102             ELSE
103               DZB=(THS(II,JJ+1)-THS(II,JJ-1))/(2.*DYT)
104             ENDIF
105           ENDIF
106     C
107           RETURN
108           END
```

```
001     C  SUBROUTINE POLUT.UPDATE
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO UPDATE THE PUFF CHARACTERISTICS
005     C  AFTER A MOVE.
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE UPDATE (N,NG,NS,ICONFG,DELTAT,NSGT)
010     C
011           PARAMETER
012          +    NPU=100,       NSC=5,         NSG=12
013     C
014           COMMON /PUFF1/
015          +   A(NPU,NSG,NSC),     B(NPU,NSG,NSC),     IPCOD(NPU,NSG,NSC),
016          +   SGPRY(NPU,NSC),     SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
017          +   SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
018          +   SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
019          +   TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),     ZMIX(NPU,NSG,NSC),
020          +   QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
021          +   NPUFFV(NSC)
022     C
023           COMMON /TPDT/
024          +    ANQ(2),          BNQ(2),          DSMQ(2),        HTMIXQ(2),
025          +    IPCODQ(2),       SYDPQ(2),        SYDGQ(2),       SYSHQ(2),
026          +    SYTOQ(2),        SZDQ(2),         SZTOQ(2),       ZNQ(2),
027          +    ANQS(2),         BNQS(2),         DSMQS(2),       ZNQS(2),
028          +    AF(4),           BF(4),           QF(2)
029     C
030           IF (ICONFG.EQ.3)  THEN
031             K=N-1
032             J=2
033             I=1
034           ELSE
035             K=N
036             I=ICONFG
037             J=ICONFG
038           ENDIF
039     C
040           IF(NG.EQ.NSGT+1) THEN
041             A(K,NG+1,NS)=ANQS(I)
042             B(K,NG+1,NS)=BNQS(I)
043           ENDIF
044           A(K,NG,NS)=ANQ(I)
045           B(K,NG,NS)=BNQ(I)
046           Z(K,NG,NS)=ZNQ(I)
047           SGTOY(K,NG,NS)=SYTOQ(I)
048           SGDPY(K,NG,NS)=SYDPQ(I)
049           SGDGY(K,NG,NS)=SYDGQ(I)
050           SGSHY(K,NG,NS)=SYSHQ(I)
051           SGTOZ(K,NG,NS)=SZTOQ(I)
052           SGDZ(K,NG,NS)=SZDQ(I)
053           ZMIX(K,NG,NS)=HTMIXQ(I)
054           IPCOD(K,NG,NS)=IPCODQ(I)
055           TIME(K,NG,NS)=TIME(K,NG,NS)+DELTAT
056           STOTM(K,NG,NS)=STOTM(K,NG,NS)+DSMQ(I)
057     C
058           ANQ(I)=ANQ(J)
059           BNQ(I)=BNQ(J)
060           ZNQ(I)=ZNQ(J)
061           ANQS(I)=ANQS(J)
062           BNQS(I)=BNQS(J)
```

```
063             HTMIXQ(I)=HTMIXQ(J)
064             IPCODQ(I)=IPCODQ(J)
065             SYTOQ(I)=SYTOQ(J)
066             SYDPQ(I)=SYDPQ(J)
067             SYDGQ(I)=SYDGQ(J)
068             SYSHQ(I)=SYSHQ(J)
069             SZTOQ(I)=SZTOQ(J)
070             SZDQ(I)=SZDQ(J)
071             DSMQ(I)=DSMQ(J)
072     C
073             RETURN
074             END
```

```
001     C  SUBROUTINE: POLUT.UTM
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO CONVERT FROM UNIVERSAL TRANSVERSE
005     C  MERCATOR (UTM) COORDINATES IN KILOMETERS TO GRAMA COORDINATES IN
006     C  KILOMETERS.  THE GRAMA GRID IS CONTAINED WITHIN UTM GRID ZONES
007     C  12 AND 13.  EQUATIONS FOR A TRANSVERSE MERCATOR PROJECTION
008     C  ASSUMING THE EARTH IS SHAPED AS A SPHERE ARE USED.  THIS IS AN
009     C  APPROXIMATION TO THE UTM GRID IN WHICH THE TRANSFORMATION
010     C  EQUATIONS ARE BASED ON THE EARTH SHAPED AS AN ELLIPSOID.
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE INPUTS ARE:
013     C       IZONE = 2  (UTM GRID ZONE 12) OR,
014     C               3  (UTM GRID ZONE 13).
015     C       UTME = UTM EASTING COORDINATE (KM)
016     C       UTMN = UTM NORTHING COORDINATE (KM)
017     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
018     C  THE OUTPUTS ARE:
019     C       COX = X-COORDINATE IN GRAMA GRID SYSTEM (KM)
020     C       COY = Y-COORDINATE IN GRAMA GRID SYSTEM (KM)
021     C  CONSTANT DEFINITIONS:
022     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
023     C       RAD = 6367.3858 [MEAN RADIUS OF EARTH (KM)]
024     C       C = .9996 [PROJECTION SCALING FACTOR]
025     C       CM = 111. [CENTRAL MERIDIAN OF UTM ZONE 13 (DEG)]
026     C          = 105. [CENTRAL MERIDIAN OF UTM ZONE 12 (DEG)]
027     C       OX = 499.472 [ADJ. ORIG. OF ZONE 13 WEST OF CM (KM)]
028     C          = 500.264 [ADJ. ORIG. OF ZONE 12 WEST OF CM (KM)]
029     C       OY = 15.943 [ADJ. ORIG. OF ZONE 13 NORTH OF EQTR. (KM)]
030     C          = 15.947 [ADJ. ORIG. OF ZONE 12 NORTH OF EQTR. (KM)]
031     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
032     C
033             SUBROUTINE UTM (IZONE,UTME,UTMN,COX,COY)
034     C
035             DOUBLE PRECISION B,XPRIM,YPRIM,A,D
036     C
037     C                    ***CONSTANTS
038             CO1=180./3.141592654
039             CO2=3.141592654/2.
040     C
041             IF (IZONE.EQ.2) GO TO 1
042     C                  ***DEFINE CONSTANT PARAMETERS FOR ZONE 13.
043             R=6367.3858
044             C=.9996
045             CM=105.
046             X0=499.472
047             Y0=15.943
048             GO TO 2
049     C                  ***DEFINE CONSTANT PARAMETERS FOR ZONE 12.
050           1 CONTINUE
051             R=6367.3858
052             C=.9996
053             CM=111.
054             X0=500.264
055             Y0=15.947
056     C                  ***COMPUTE LATITUDE-LONGITUDE IN DEGREES.
057           2 CONTINUE
058             B=EXP(2.*(UTME-X0)/(C*R))
059             XPRIM=(B-1.)/(B+1.)
060             YPRIM=TAN(CO2-(UTMN+Y0)/(C*R))
061             A=DSQRT((1.-XPRIM**2)/(1.+YPRIM**2))
062             D=DACOS(YPRIM*A/DSQRT(1.-A**2))*CO1
```

```
063           XLAT=DASIN(A)*CO1
064           IF(XPRIM.LT.0.) THEN
065             YLONG=CM+D
066           ELSE
067             YLONG=CM-D
068           ENDIF
069     C
070                 ***CONVERT LATITUDE-LONGITUDE TO GRAMA COORDINATES.
071           CALL LAT (XLAT,YLONG,COX,COY)
072     C
073           RETURN
              END
```

```
001     C  SUBROUTINE POLUT.VALMA
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE DETERMINES TOTAL POLLUTANT MASS IN EACH VALLEY
005     C  SEGMENT WHEN COUPLING STARTS.
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE VALMA(NS,CLK,CLKD,NSGA,NPLT)
010     C
011           PARAMETER
012          +     NSG=12,        NSC=5,          NVA=5
013     C
014           COMMON /VALLY1/
015          +   COEF(NSG,NVA,3),COXV(NVA),      COYV(NVA),      WSV(NVA),
016          +   SEGX(NSG,NVA),  SEGY(NSG,NVA), NSEG(NVA),      IDVAL(NVA),
017          +   NVALE,          QM(NSG,NSC,2), VSEGL(NSG,NVA),WIDV(NSG,NVA)
018     C
019           COMMON /SOURC1/
020          +    EMIS(NSC,2),   NSOURC,         RS(NSC),        TS(NSC),
021          +    VS(NSC),       XSL(NSC),       YSL(NSC),       ZSH(NSC),
022          +    ELS(NSC),      AS(NSC),        BS(NSC),        SIGIN(NSC),
023          +    IDSVAL(NSC),   VALL(NSC),      INDXV(NSC)
024     C
025           IV=INDXV(NS)
026           DUR=(CLK-CLKD)*3600.
027           DIST=WSV(IV)*DUR/1000.
028           DISTC=0.
029           JJ=0
030           DO 10 J=NSEG(IV),1,-1
031             JJ=JJ+1
032             DISTC=DISTC+VSEGL(J,IV)
033             IF(DISTC.GT.DIST) THEN
034               AA=DIST-(DISTC-VSEGL(J,IV))
035               CK=AA/VSEGL(J,IV)
036               IF(CK.GT..5) THEN
037                 NSGA=JJ
038               ELSE
039                 NSGA=JJ-1
040               ENDIF
041               GO TO 20
042             ENDIF
043          10 CONTINUE
044             NSGA=NSEG(IV)
045          20 CONTINUE
046             TOT=0.
047             NGB=NSEG(IV)-NSGA+1
048             DO 15 J=NGB,NSEG(IV)
049               TOT=VSEGL(J,IV)+TOT
050          15 CONTINUE
051             DO 40 L=1,NPLT
052               RMASS=DUR*EMIS(NS,L)
053               DO 30 J=NGB,NSEG(IV)
054                 FRAC=VSEGL(J,IV)/TOT
055                 QM(J,NS,L)=RMASS*FRAC
056          30   CONTINUE
057          40 CONTINUE
058     C
059             RETURN
060             END
```

```
C  SUBROUTINE  POLUT.VALY
C  DATE: APRIL 1985
C
C    THIS PROGRAM IS USED TO COMPUTE POLLUTANT CONCENTRATIONS FROM
C  VENTILATION FROM UP TO 5 VALLEYS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE VALY(CLK,DELTAT,NSGA,NS,NHC,CONT,FLAG,IPRR,NPLT)
C
      PARAMETER
     +    NPU=100,      NSC=5,        NVA=5,        NSG=12
C
      LOGICAL
     +    LUFORM,       WOFF,         VALL,          CONT,
     +    FLAG
C
      COMMON /MAIN2/
     +    HA,           HH,           IHH,          IXO,
     +    IYO,          ROA,          RXY,          SLX,
     +    SLY
C
      COMMON /VALLY1/
     +    COEF(NSG,NVA,3),COXV(NVA),      COYV(NVA),     WSV(NVA),
     +    SEGX(NSG,NVA),  SEGY(NSG,NVA), NSEG(NVA),     IDVAL(NVA),
     +    NVALE,          QM(NSG,NSC,2), VSEGL(NSG,NVA),WIDV(NSG,NVA)
C
      COMMON /PUFF1/
     +    A(NPU,NSG,NSC),       B(NPU,NSG,NSC),       IPCOD(NPU,NSG,NSC),
     +    SGPRY(NPU,NSC),       SGDPY(NPU,NSG,NSC), SGDGY(NPU,NSG,NSC),
     +    SGSHY(NPU,NSG,NSC), SGTOY(NPU,NSG,NSC), SGPRZ(NPU,NSC),
     +    SGDZ(NPU,NSG,NSC),  SGTOZ(NPU,NSG,NSC), STOTM(NPU,NSG,NSC),
     +    TIME(NPU,NSG,NSC),  Z(NPU,NSG,NSC),       ZMIX(NPU,NSG,NSC),
     +    QTP(NPU,NSG,NSC,2), IQCOD(NPU,NSG,NSC), NPUFFS(NSC),
     +    NPUFFV(NSC)
C
      COMMON /TPDT/
     +    ANQ(2),       BNQ(2),       DSMQ(2),      HTMIXQ(2),
     +    IPCODQ(2),    SYDPQ(2),     SYDGQ(2),     SYSHQ(2),
     +    SYTOQ(2),     SZDQ(2),      SZTOQ(2),     ZNQ(2),
     +    ANQS(2),      BNQS(2),      DSMQS(2),     ZNQS(2),
     +    AF(4),        BF(4),        QF(2)
C
      COMMON /RUNSP/
     +    LUFORM,       METHOD,       WOFF
C
      COMMON /SOURC1/
     +    EMIS(NSC,2),  NSOURC,       RS(NSC),      TS(NSC),
     +    VS(NSC),      XSL(NSC),     YSL(NSC),     ZSH(NSC),
     +    ELS(NSC),     AS(NSC),      BS(NSC),      SIGIN(NSC),
     +    IDSVAL(NSC),  VALL(NSC),    INDXV(NSC)
C
      RIXO=REAL(IXO)
      RIYO=REAL(IYO)
      RIPRR=REAL(IPRR)
C
      IV=INDXV(NS)
      NSGT=NSEG(IV)
      NGB=NSGT-NSGA+2
      IF(CONT) THEN
        NPUFFV(NS)=NPUFFV(NS)+1
        N=NPUFFV(NS)
```

```
063             DO 16 NG=2,NGB-1
064               IQCOD(N,NG,NS)=4
065          16   CONTINUE
066             DO 18 NG=NGB,NSGT+1
067               IQCOD(N,NG,NS)=5
068          18   CONTINUE
069             GO TO 20
070           ELSE
071             IF(FLAG) THEN
072               FLAG=.FALSE.
073               NPUFFV(NS)=NPUFFV(NS)+1
074               N=NPUFFV(NS)
075               DO 17 NG=2,NGB-1
076                 IQCOD(N,NG,NS)=4
077          17     CONTINUE
078               DO 19 NG=NGB,NSGT+1
079                 IQCOD(N,NG,NS)=2
080          19     CONTINUE
081               GO TO 20
082             ENDIF
083             N=NPUFFV(NS)
084             IF(N.EQ.0) RETURN
085             GO TO 21
086           ENDIF
087     C
088     C               ***INITIALIZE CHARACTERISTICS OF STARTING PUFF FROM EACH
089     C               ***SOURCE.
090          20 CONTINUE
091           DO 38 NG=NGB,NSGT+1
092           A(N,NG,NS)=(SEGX(NG-1,IV)-RIX0)/SLX
093           B(N,NG,NS)=(SEGY(NG-1,IV)-RIY0)/SLY
094           TIME(N,NG,NS)=0.
095           STOTM(N,NG,NS)=1.
096           IPCOD(N,NG,NS)=1
097           Z(N,NG,NS)=5.
098           DO 28 L=1,NPLT
099             QTP(N,NG,NS,L)=QM(NG-1,NS,L)*COEF(NG-1,IV,NHC)/RIPRR
100             IF(COEF(NG-1,IV,NHC).EQ.1.) QTP(N,NG,NS,L)=0.
101          28 CONTINUE
102           CALL BLPINT (A(N,NG,NS),B(N,NG,NS),CLK,HMIX,ISTB,USTR,CAPL.WSTR,
103          +             TSURF,PRESS)
104           ZSHP=Z(N,NG,NS)
105           ZMIX(N,NG,NS)=HMIX
106           CALL WINDS (A(N,NG,NS),B(N,NG,NS),ZSHP,CLK,U,V,W,WT)
107           SPEED=SQRT(U**2+V**2)
108           IF (Z(N,NG,NS).GE.ZMIX(N,NG,NS)) IPCOD(N,NG,NS)=3
109           IF (LUFORM.AND.IPCOD(N,NG,NS).EQ.1) IPCOD(N,NG,NS)=2
110           CALL YINITV (WIDV(NG-1,IV),SGDPY(N,NG,NS),SGDGY(N,NG,NS),
111          +             SGSHY(N,NG,NS),SGTOY(N,NG,NS))
112           CALL ZINITV (WIDV(NG-1,IV),METHOD,USTR,WSTR,HMIX,CAPL.ISTB,
113          +             A(N,NG,NS),B(N,NG,NS),ZSHP,SPEED,SGDZ(N,NG,NS),
114          +             SGTOZ(N,NG,NS))
115          38 CONTINUE
116           A(N,NSGT+2,NS)=AS(NS)
117           B(N,NSGT+2,NS)=BS(NS)
118     C
119          21 DO 48 NG=2,NSGT+1
120     C               ***COMPUTE CONCENTRATION FOR LAST PUFF.
121           N=1
122          29 IF (IQCOD(N,NG,NS).EQ.4)  THEN
123             N=N+1
124             IF (N.GT.NPUFFV(NS)) GO TO 33
```

```
125           GO TO 29
126          ENDIF
127          CALL NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD(N,NG,NS),ZSHP,ELS(NS),
128         +             A(N,NG,NS),B(N,NG,NS),Z(N,NG,NS),CLK,ANQ(1),BNQ(1),
129         +             ZNQ(1),IQCOD(N,NG,NS),DSMQ(1))
130          CALL NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD(N,NG,NS),ZSHP,ELS(NS),
131         +             A(N,NG+1,NS),B(N,NG+1,NS),Z(N,NG+1,NS),CLK,ANQS(1),
132         +             BNQS(1),ZNQS(1),IQCOD(N,NG,NS),DSMQS(1))
133          IF (IQCOD(N,NG,NS).EQ.4)  THEN
134            N=N+1
135            IF (N.GT.NPUFFV(NS)) GO TO 33
136            IF (IQCOD(N,NG,NS).EQ.1) IQCOD(N,NG,NS)=0
137            IF (IQCOD(N,NG,NS).EQ.2) IQCOD(N,NG,NS)=4
138            GO TO 29
139          ENDIF
140     C              CALCULATE SIGMAY AND SIGMAZ
141          CALL SIGGS(N,NG,NS,CLK,DELTAT,1)
142     C
143          IF (IQCOD(N,NG,NS).EQ.5)  THEN
144     C          ***FIRST PUFF PROCESSING
145            ICONFG=1
146            CALL CALCVA(N,NG,NS,ICONFG,NPLT,NSGT,DELTAT)
147            IQCOD(N,NG,NS)=0
148            GO TO 33
149          ENDIF
150     C
151     C             ***COMPUTE CONCENTRATION FOR REST OF PUFFS.
152        31 N=N+1
153          CALL NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD(N,NG,NS),ZSHP,ELS(NS),
154         +             A(N,NG,NS),B(N,NG,NS),Z(N,NG,NS),CLK,ANQ(2),BNQ(2),
155         +             ZNQ(2),IQCOD(N,NG,NS),DSMQ(2))
156          CALL NEWLOC (WOFF,DELTAT,SLX,SLY,IPCOD(N,NG,NS),ZSHP,ELS(NS),
157         +             A(N,NG+1,NS),B(N,NG+1,NS),Z(N,NG+1,NS),CLK,ANQS(2),
158         +             BNQS(2),ZNQS(2),IQCOD(N,NG,NS),DSMQS(2))
159     C
160          IF (IQCOD(N,NG,NS).EQ.4)  THEN
161            N=N-1
162            IF (IQCOD(N,NG,NS).EQ.0) IQCOD(N,NG,NS)=4
163            IF (IQCOD(N,NG,NS).EQ.1) IQCOD(N,NG,NS)=2
164            ICONFG=1
165            CALL UPDATE(N,NG,NS,ICONFG,DELTAT,NSGT)
166            N=N+2
167            IF(N.GT.NPUFFV(NS)) GO TO 33
168            IF (IQCOD(N,NG,NS).EQ.1) IQCOD(N,NG,NS)=0
169            IF (IQCOD(N,NG,NS).EQ.2) IQCOD(N,NG,NS)=4
170            GO TO 29
171          ENDIF
172     C             ***CALCULATE SIGMAY AND SIGMAZ
173          CALL SIGGS (N,NG,NS,CLK,DELTAT,2)
174     C
175     C             ***COMPUTING CONC FOR QUADRILATERAL
176          ICONFG=3
177          CALL CALCVA(N,NG,NS,ICONFG,NPLT,NSGT,DELTAT)
178     C
179          IF (IQCOD(N,NG,NS).EQ.5)  THEN
180     C             ***COMPUTING CONC FOR LAST PUFF FOR TIME STEP
181            ICONFG=2
182            CALL CALCVA(N,NG,NS,ICONFG,NPLT,NSGT,DELTAT)
183            IQCOD(N,NG,NS)=1
184            GO TO 33
185          ENDIF
186          IF (IQCOD(N,NG,NS).EQ.2)  THEN
```

```
187             ICONFG=2
188             CALL "PDATE(N,NG,NS,ICONFG,DELTAT,NSGT)
189             N=N+1
190             IF (N.GT.NPUFFV(NS)) GO TO 33
191             GO TO 29
192           ENDIF
193           GO TO 31
194     C
195        33 CONTINUE
196        48 CONTINUE
197     C
198           RETURN
199           END
```

```
001     C  SUBROUTINE: POLUT.VERTDF
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO CALCULATE THE VERTICAL DISTRIBUTION
005     C  FUNCTION. IT IS TAKEN FROM THE MESOPUFF MODEL (ERT).
006     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
007     C  THE INPUTS ARE:
008     C      SIGZ - SIGMA-Z (M)
009     C      PHT  - PLUME HEIGHT  (M)
010     C      EL   - LID HEIGHT  (M)
011     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
012     C  THE OUTPUT IS:
013     C      VDF  - VERTICAL DIFFUSION EXPONENTIAL
014     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
015     C
016           SUBROUTINE VERTDF (SIGZ,PHT,EL,VDF)
017
018           THT=0.0
019           TERCOR=1.0
020           TERAN=(1.-TERCOR)*AMIN1(THT,PHT)
021           ELT=EL-TERAN
022           HPT=PHT-TERAN
023           H=HPT/ELT
024           T=(SIGZ/ELT)**2
025     C               ***THESE VALUES OF TMIN & TMAX GIVE 2 DECIMAL ACCURACY
026           TMIN=.1
027           TMAX=7.0
028           PISQ=9.86960
029           PI=3.14159
030           TWOPI=6.28319
031           SUM=1.
032           IF (T.GE..6) GO TO 2
033           ARG=2*(1-H)/T
034           IF (ARG.GE.TMAX) GO TO 1
035           IF (ARG.LT.TMIN) SUM=SUM+1-ARG
036           IF (ARG.GE.TMIN) SUM=SUM+EXP(-ARG)
037           ARG=2*(1+H)/T
038           IF (ARG.GE.TMAX) GO TO 1
039           SUM=SUM+EXP(-ARG)
040           ARG=4*(2-H)/T
041           IF (ARG.GE.TMAX) GO TO 1
042           SUM=SUM+EXP(-ARG)
043           ARG=4*(2+H)/T
044           IF (ARG.LT.TMAX) SUM=SUM+EXP(-ARG)
045         1 ARG=-.5*H**2/T
046           IF (ARG.LT.-90.) VDF=0.
047           IF (ARG.GE.-90.) VDF=2.*SUM*EXP(ARG)
048           GO TO 4
049     C
050         2 ARG=PISQ*T/2
051           IF (ARG.GE.TMAX) GO TO 3
052           SUM=SUM+2*EXP(-ARG)*COS(PI*H)
053           ARG=2*PISQ*T
054           IF (ARG.LT.TMAX) SUM=SUM+2*EXP(-ARG)*COS(TWOPI*H)
055         3 VDF=SUM*SQRT(TWOPI)*SIGZ/ELT
056     C
057         4 RETURN
058           END
```

```
001     C  SUBROUTINE: POLUT.VREAL
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE THE UNTRANSFORMED VELOCITY
005     C  COMPONENTS FOR ANY ONE POINT ON A GIVEN 'GAMMA' SURFACE.
006     C  EACH OF THE THREE VELOCITY COMPONENTS ARE FOR THE 'X', 'Y'
007     C  AND 'GAMMA' COORDINATES.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C    THRU ARGUMENT:
011     C      FI1 = ARRAY OF AMPLITUDE FUNCTION 'FSUBI' FOR A GAMMA SURFACE
012     C      GI1 = ARRAY OF AMPLITUDE FUNCTION 'GSUBI' FOR A GAMMA SURFACE
013     C      GMF1 = ARRAY OF AMPL FNCT 'GSUBI-FSUBI' FOR A GAMMA SURFACE
014     C      BI1 = ARRAY OF POLY BASIS FUNCTIONS 'BSUBI' FOR A GAMMA SRFC
015     C      CI1 = ARRAY OF POLY BASIS FUNCTIONS 'CSUBI' FOR A GAMMA SRFC
016     C      PBI1 = ARRAY OF POLY BASIS FNCTS 'PARTIALB' FOR A GAMMA SRFC
017     C      RRO = AIR DENSITY AT POINT (KG/M**3)
018     C      DZA = TERRAIN SLOPE IN ALPHA-DIRECTION (M)
019     C      DZB = TERRAIN SLOPE IN BETA-DIRECTION (M)
020     C      NTERMS = NUMBER OF POLYNOMIAL TERMS DESCRIBING WINDS
021     C      PSF = GAMMA-COORDINATE OF POINT
022     C      DHA = SLOPE IN ALPHA-DIRECTION OF TOP BOUNDARY (M)
023     C      DHB = SLOPE IN BETA-DIRECTION OF TOP BOUNDARY (M)
024     C      HZS = 'SMALL H' (M)
025     C      SLX = DOMAIN X-DIMENSION (KM)
026     C      RXY = RATIO OF DOMAIN X-DIMENSION TO Y-DIMENSION
027     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
028     C  THE OUTPUTS ARE:
029     C    THRU ARGUMENT:
030     C      RU1 = X-COMPONENT OF VELOCITY (M/S)
031     C      RV1 = Y-COMPONENT OF VELOCITY (M/S)
032     C      RW1 = GAMMA-COMPONENT OF VELOCITY (/S)
033     C      RWT = GAMMA-COMP OF VELOCITY SUFFICIENT TO FOLLOW TERRAIN (/S)
034     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
035     C  PARAMETERS DEFINITIONS:
036     C      NPO = NUMBER OF POLYNOMIAL TERMS
037     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
038     C
039           SUBROUTINE VREAL (FI1,GI1,GMF1,BI1,CI1,PBI1,RRO,DZA,DZB,NTERMS.
040          +                  PSF,DHA,DHB,HZS,SLX,RXY,RU1,RV1,RW1,RWT)
041     C
042           PARAMETER
043          +    NPO=21
044     C
045           DIMENSION
046          +    BI1(NPO),      CI1(NPO),      FI1(NPO),      GI1(NPO),
047          +    GMF1(NPO),     PBI1(NPO)
048     C
049     C
050           SL=SLX*1000.
051     C               **COMPUTE BIG U, BIG V AND BIG W.
052           U=0.0
053           V=0.0
054           W=0.0
055     C
056           DO 1 J=1,NTERMS
057             U=FI1(J)*BI1(J)+U
058             V=GI1(J)*CI1(J)+V
059             W=GMF1(J)*PBI1(J)+W
060         1 CONTINUE
061     C               **COMPUTE REAL VELOCITY COMPONENTS.
062           RU1=U/(RRO*HZS)
```

```
063           RV1=V/(RXY*RRO*HZS)
064           ZDS=DZA*RU1+RXY*DZB*RV1
065           HD=DHA*RU1+RXY*DHB*RV1
066           RW1=(W+RRO*ZDS+RRO*PSF*HD)/(RRO*SL)
067     C               ***COMPUTE VERTICAL VELOCITY SUFFICIENT TO FOLLOW TERRAIN.
068           RWT=(ZDS+PSF*HD)/SL
069     C
070           RETURN
071           END
```

```
C  SUBROUTINE POLUT.WEIGHT
C  APRIL 1985
C
C    THIS SUBROUTINE COMPUTES AN ARRAY OF WEIGHTING FACTORS USED
C  TO SPLIT PROPERTIES OF PLUME SEGMENTS AND QUADRILATERALS.  IT
C  IS BASED ON PRINCIPLE THAT PUFFS NOT BE SEPARATED BY MORE THAN
C  ONE SIGMAY.
C
C  THE INPUTS ARE:
C       D = DISTANCE BETWEEN END PUFFS (M)
C       RE = SIGMAY OF ENDING PUFF (M)
C       RB = SIGMAY OF BEGINING PUFF (M)
C
C  THE OUTPUTS ARE:
C       NP = NUMBER OF POINTS INTERNAL ON SEGMENT
C       WF = ARRAY OF WEIGHTING FACTORS.  THE SUM OF THE FACTORS IS 1.0.
C            THE FACTOR IS THE RATIO OF THE DISTANCES BETWEEN PUFFS TO
C            THE SEGMENT LENGTH.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE WEIGHT(D,RB,RE,NP,WF)
C
      PARAMETER
     +    IP=50
C
      DIMENSION
     +    WF(IP)
C
      IF(D.LE.0.) THEN
        NP=1
        WF(1)=1.
        RETURN
      ENDIF
      NPMAX=IP-1
      I=0
      C=0.
      WFP=0.
      DR=RE-RB
    3 I=I+1
      IF(I.EQ.NPMAX) THEN
        WF(I)=1.-WFP
        NP=I
        RETURN
      ENDIF
      RI=(D*RE-C*DR)/(D+DR)
      C=C+RI
      IF(C.LT.D) THEN
        WF(I)=RI/D
        WFP=WFP+WF(I)
        GO TO 3
      ELSE
        WF(I)=1.-WFP
        NP=I
      ENDIF
C
      RETURN
      END
```

```
001     C  SUBROUTINE POLUT.WINDS
002     C  DATE: APRIL 1985
003     C
004     C     THIS SUBROUTINE IS USED TO COMPUTE THE WINDS AT A POINT
005     C  GIVEN THE LOCATION OF THE POINT, THE AMPLITUDE FUNCTIONS FOR THREE
006     C  TIMES AND THE TERRAIN DATA.
007     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
008     C     THE INPUTS ARE:
009     C       A - ALPHA COORDINATE OF POINT TO COMPUTE WINDS.
010     C       B - BETA COORDINATE OF POINT.
011     C       Z - VERTICAL COORDINATE OF POINT (M AGL).
012     C       CLK - TIME AT WHICH WINDS ARE TO BE COMPUTED (HR OF YEAR).
013     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
014     C     THE OUTPUTS ARE:
015     C       U - COMPONENT OF WINDS IN X-DIRECTION (M/SEC).
016     C       V - COMPONENT OF WINDS IN Y-DIRECTION (M/SEC).
017     C       W - COMPONENT OF WINDS IN Z-DIRECTION (M/SEC).
018     C       WT - COMPONENT OF WINDS IN Z-DIRECTION SUFFICIENT ONLY
019     C            TO FOLLOW TERRAIN (M/SEC).
020     C
021     C     THE AMPLITUDE FUNCTIONS FOR EACH GAMMA SURFACE FOR THE THREE
022     C     TIMES ARE PASSED THRU COMMON BLOCK 'WINDS1'.
023     C
024     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
025     C
026           SUBROUTINE WINDS (A,B,Z,CLK,U,V,W,WT)
027     C
028           PARAMETER
029          +     NGA=9,          NPO=21
030     C
031           DIMENSION
032          +     BI(NPO),        CI(NPO),        FI1(NPO),       GI1(NPO),
033          +     GMF1(NPO),      LA(2),          PBI(NPO),       RU(4),
034          +     RV(4),          RW(4),          RWT(4)
035     C
036           COMMON /MAIN2/
037          +     HA,             HH,             IHH,            IXO,
038          +     IYO,            ROA,            RXY,            SLX,
039          +     SLY
040     C
041           COMMON /CLOCK1/
042          +     CLK1,           CLK2,           CLK3
043     C
044           COMMON /WINDS1/
045          +     FI(NPO,NGA,3), GI(NPO,NGA,3), GMF(NPO,NGA,3), HTGAMA(NGA),
046          +     THETTA(3),      NGAMA,          NTERMS(3)
047     C
048           CALL TETRI (A,B,TH,DZA,DZB)
049           CALL SHSLP (IHH,DZA,DZB,A,B,DHA,DHB)
050           CALL SMALLH (IHH,HH,TH,A,B,HZS)
051     C
052     C              ***DEFINE TWO GAMMA SURFACES TO INTERPOLATE WINDS
053     C              ***BETWEEN.
054           G=Z/HZS
055           IF (G.GT.1.0) G=1.
056           IF (G.LT.HTGAMA(1)) G=HTGAMA(1)
057           IF (G.EQ.HTGAMA(NGAMA)) GO TO 3
058           IF (G.EQ.HTGAMA(1)) GO TO 4
059           DO 1 K=NGAMA-1,1,-1
060           IF (G.GE.HTGAMA(K)) GO TO 2
061         1 CONTINUE
062           WRITE (6,11)
```

```
063           STOP 'WINDS'
064         2 LA(1)=K
065           LA(2)=K+1
066           GO TO 5
067         3 LA(1)=NGAMA
068           LA(2)=NGAMA
069           GO TO 5
070         4 LA(1)=1
071           LA(2)=1
072           GO TO 5
073         5 CONTINUE
074     C
075     C               ***COMPUTE WIND COMPONENTS FOR TWO SURFACES FOR TWO TIMES.
076           NS=0
077           NTS=0
078           IF (CLK.GE.CLK2) NTS=1
079           DO 7 NT=1+NTS,2+NTS
080           CALL POLBAS (A,B,NTERMS(NT),BI,CI,PBI)
081           DO 7 NL=1,2
082           NS=NS+1
083           ZL=TH+HTGAMA(LA(NL))*HZS
084           ROA=353.1/THETTA(NT)
085           HA=102.4*THETTA(NT)
086           RO=ROA*(1.-ZL/HA)**2.5
087           DO 6 NN=1,NTERMS(NT)
088           FI1(NN)=FI(NN,LA(NL),NT)
089           GI1(NN)=GI(NN,LA(NL),NT)
090           GMF1(NN)=GMF(NN,LA(NL),NT)
091         6 CONTINUE
092           CALL VREAL (FI1,GI1,GMF1,BI,CI,PBI,RO,DZA,DZB,NTERMS(NT),HTGAMA(LA
093          +(NL)),DHA,DHB,HZS,SLX,RXY,RU(NS),RV(NS),RW(NS),RWT(NS))
094         7 CONTINUE
095     C               ***SPATIALLY AND TEMPORALLY INTERPOLATE WINDS.
096           IF (LA(2).EQ.LA(1)) GO TO 8
097           S2=(G-HTGAMA(LA(1)))/(HTGAMA(LA(2))-HTGAMA(LA(1)))
098           S1=1.-S2
099           GO TO 9
100         8 S2=1.
101           S1=0.
102         9 CONTINUE
103           T2=(CLK-CLK1)/(CLK2-CLK1)
104           IF (CLK.GE.CLK2) T2=ABS((CLK-CLK2)/(CLK3-CLK2))
105           T1=1.-T2
106           U=T2*(S2*RU(4)+S1*RU(3))+T1*(S2*RU(2)+S1*RU(1))
107           V=T2*(S2*RV(4)+S1*RV(3))+T1*(S2*RV(2)+S1*RV(1))
108           W=T2*(S2*RW(4)+S1*RW(3))+T1*(S2*RW(2)+S1*RW(1))
109           WT=T2*(S2*RWT(4)+S1*RWT(3))+T1*(S2*RWT(2)+S1*RWT(1))
110           IF(U.EQ.0..AND.V.EQ.0.) U=.1
111     C
112           RETURN
113        11 FORMAT (5X,' GAMMA OF POINT NOT WINTIN GAMMA DOMAIN IN',' SUBROUTI
114          +NE WINDS.')
115           END
```

```
001     C  SUBROUTINE POLUT.YINIT
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO SPECIFY THE INITIAL CONDITIONS FOR THE
005     C  COMPONENTS OF SIGMAY.  INITIAL SPREAD DUE TO PLUME RISE, ATMOSPHERIC
006     C  TURBULENCE, AND WIND SHEAR ARE SPECIFIED.
007     C
008     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
009     C
010           SUBROUTINE YINIT (SYAREA,PR,SYPR,SYDP,SYDG,SYSH,SYTOT)
011     C
012           SYPR=PR/SQRT(10.)
013           SYDP=SYAREA
014           IF(SYDP.EQ.0.) SYDP=50.
015           SYDG=20.
016           SYSH=0.
017           SYTOT=SQRT(SYPR**2+SYDP**2)
018           RETURN
019           END
```

```
C  SUBROUTINE POLUT.YINITV
C  APRIL 1985
C
C    THIS SUBROUTINE IS USED TO SPECIFY THE INITIAL CONDITIONS FOR THE
C  COMPONENTS OF SIGMAY FROM A VALLEY RELEASE.  INITIAL SPREAD DUE TO
C  ATMOSPHERIC TURBULENCE AND WIND SHEAR ARE SPECIFIED.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE YINITV (WIDV,SYDP,SYDG,SYSH,SYTOT)
C
      SYDP=WIDV/2.
      IF(SYDP.LT.500.) SYDP=500.
      SYDG=50.
      SYSH=0.
      SYTOT=SYDP
      RETURN
      END
```

```
001     C  SUBROUTINE POLUT.YVIRT
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE VIRTUAL TIME FOR THE
005     C  COMPUTATION OF PLUME SIGMA Y DUE TO ATMOSPHERIC TURBULENCE
006     C  USING PASQUILL'S GENERAL FORM.
007     C  VIRTUAL TIME IS THE ELAPSED TIME A PLUME WOULD HAVE HAD TO TRAVEL
008     C  UNDER THE STEADY-STATE CONDITIONS AT T+DELTAT (NEW TIME) TO BE
009     C  THE SIZE OF THE PUFF AT TIME T (OLD TIME).
010     C      THE INPUTS ARE:
011     C        TIM = INITIAL GUESS AT DIFF. TIME WHICH RESULTS IN BY (S)
012     C              MAKE INITIAL GUESS TIME T (OLD TIME)
013     C        IS = P-G-T STAB. (1,2,3,4,5, OR 6) AT T+DELTAT (NEW TIME)
014     C        SIGV = STD. DEV. IN HORIZONTAL WINDS AT T+DELTAT (M/S)
015     C        BY = HORIZONTAL TURB. DIFF. AT TIME T (OLD TIME)  (M)
016     C        HAG = HEIGHT OF PUFF AT T+DELTAT (M AGL)
017     C      THE OUTPUT IS:
018     C        VT = VIRTUAL TIME (S)
019     C
020     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
021     C
022           SUBROUTINE YVIRT (TIM,SIGV,BY,VT)
023     C
024     C               ***TREATMENT FOR FIRST TIME STEP (OLD TIME T=0)
025           IF(TIM.EQ.0..AND.BY.EQ.0.) THEN
026             VT=0.
027             RETURN
028           ENDIF
029           IF(TIM.EQ.0..AND.BY.NE.0.) THEN
030             TIM=500.
031           ENDIF
032     C
033     C               ***COMPUTE VIRTUAL TIME USING NEWTONS METHOD.
034           C3=1000.
035           IF(BY.LE.0.)BY=1.
036           VT=TIM
037           IT=0
038         3 IT=IT+1
039           AA=SQRT(VT/C3)
040           FY=1./(1.+.9*AA)
041           FYP=-(.45*FY**2)/(C3*AA)
042           G=FY*VT-BY/SIGV
043           GP=FYP*VT+FY
044           VT=VT-G/GP
045           IF(VT.LE.0.)VT=1.
046           CK=ABS(G/GP)
047           IF (CK.LE.10.) RETURN
048           IF (IT.GT.20)  THEN
049             WRITE (6,4)
050             STOP 'YVIRT'
051           ENDIF
052           GO TO 3
053         4 FORMAT (2X,'VIRT. DIST. COMP. DID NOT CONVERGE IN 20 ITER.')
054           END
055
```

```
001     C  SUBROUTINE POLUT.YVIRTG
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE VIRTUAL TIME FOR THE
005     C  COMPUTATION OF PUFF SIGMA Y DUE TO ATMOSPHERIC TURBULENCE
006     C  USING GIFFORD'S FORMULATION.
007     C  VIRTUAL TIME IS THE ELAPSED TIME A PUFF WOULD HAVE HAD TO TRAVEL
008     C  UNDER THE STEADY-STATE CONDITIONS AT T+DELTAT (NEW TIME) TO BE
009     C  THE SIZE OF THE PUFF AT TIME T (OLD TIME).
010     C       THE INPUTS ARE:
011     C         TIM = INITIAL GUESS AT DIFF. TIME WHICH RESULTS IN BY (S)
012     C               MAKE INITIAL GUESS TIME T (OLD TIME)
013     C         SIGV = STD. DEV. IN HOR. WINDS AT T+DELTAT (NEW TIME) (M/S)
014     C         ED = LARGE SCALE EDDY DIFFUSIVITY (M**2/S)
015     C         BY = HORIZONTAL TURB. DIFF. AT TIME T (OLD TIME)  (M)
016     C       THE OUTPUT IS:
017     C         VT = VIRTUAL TIME (S)
018     C
019     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
020     C
021           SUBROUTINE YVIRTG (TIM,SIGV,ED,BY,VT)
022     C
023           DOUBLE PRECISION
024          +   SSIGV,     SED,     C1,     C2,     C22,     C3,     C4,
025          +   G,         GP,      CK,     VTT
026     C
027     C               ***TREATMENT FOR FIRST TIME STEP ( OLD TIME T=0)
028           IF(TIM.EQ.0..AND.BY.EQ.0.) THEN
029             VT=0.
030             RETURN
031           ENDIF
032             IF(TIM.EQ.0..AND.BY.NE.0.) THEN
033             TIM=1000.
034           ENDIF
035     C
036     C               ***COMPUTE VIRTUAL TIME USING NEWTONS METHOD.
037           IF(BY.LE.0.)BY=1.
038           VTT=TIM
039           SSIGV=SIGV**2
040           SED=ED**2
041           C1=2.*ED*SSIGV
042           C2=SSIGV*BY**2
043           IT=0
044         3 IT=IT+1
045           C22=C1*VTT-C2
046           C3=1./DEXP((SSIGV/ED)*VTT)
047           C4=1.-C3
048           G=C22-2.*SED*C4-SED*C4**2
049           GP=C1*(1.-C3*(2.-C3))
050           VTT=VTT-G/GP
051           IF(VTT.LE.0.)VTT=1.
052           CK=DABS(G/GP)
053           IF (CK.LE.10.) THEN
054             VT=VTT
055             RETURN
056           ENDIF
057           IF (IT.GT.20)  THEN
058             WRITE (6,4)
059             STOP 'YVIRTG'
060           ENDIF
061           GO TO 3
062         4 FORMAT (2X,'VIRT. DIST. COMP. DID NOT CONVERGE IN 20 ITER.')
```

```
063          END
064
```

```
C  SUBROUTINE POLUT.ZINIT
C  APRIL 1985
C
C    THIS SUBROUTINE IS USED TO SPECIFY THE INITIAL COMPONENTS OF
C  SIGMAZ.  THE VERTICAL SPREAD DUE TO PLUME RISE AND ATMOSPHERIC
C  TURBULENCE ARE SPECIFIED.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE ZINIT (PR,SZPR,SZD,SZTOT)
C
      SZPR=PR/SQRT(10.)
      SZD=20.
      SZTOT=SQRT(SZPR**2+SZD**2)
      RETURN
      END
```

```
C  SUBROUTINE POLUT.ZINITV
C  APRIL 1985
C
C    THIS SUBROUTINE IS USED TO SPECIFY THE INITIAL COMPONENTS OF
C  SIGMAZ FOR A VALLEY RELEASE.  THE VERTICAL SPREAD DUE TO
C  ATMOSPHERIC TURBULENCE IS SPECIFIED.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE ZINITV(WIDV,METH,USTAR,WSTAR,HMIX,OBUKV,IS,A0,B0,
     +                  HAG,SPD,SZD,SZTOT)
C
      SPEED=SPD
      IF(SPEED.LT..1) SPEED=.1
      TIM=WIDV/SPEED
C
C          ***DETERMINE STD. DEV. OF VERTICAL WINDS (SIGMA W)
      CALL SIGMAW(METH,USTAR,WSTAR,HMIX,OBUKV,IS,A0,B0,A0,B0,
     +            SPD,HAG,SIGW)
C
C          ***COMPUTE SIGMAZ USING PASQUILL'S GENERAL FORM
      IF(IS.LE.4) THEN
        C3=500.
      ELSE
        C3=50.
      ENDIF
      FZ=1./(1.+.9*SQRT(TIM/C3))
      SZD=SIGW*TIM*FZ
      SZTOT=SZD
C
      RETURN
      END
```

```
001     C  SUBROUTINE POLUT.ZVIRT
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE VIRTUAL TIME FOR THE
005     C  COMPUTATION OF PLUME SIGMA Z DUE TO ATMOSPHERIC TURBULENCE
006     C  USING PASQUILL'S GENERAL FORM.
007     C  VIRTUAL TIME IS THE ELAPSED TIME A PLUME WOULD HAVE HAD TO TRAVEL
008     C  UNDER THE STEADY-STATE CONDITIONS AT T+DELTAT (NEW TIME) TO BE
009     C  THE SIZE OF THE PUFF AT TIME T (OLD TIME).
010     C       THE INPUTS ARE:
011     C         TIM = INITIAL GUESS AT DIFF. TIME WHICH RESULTS IN BY (S)
012     C               MAKE INITIAL GUESS TIME T (OLD TIME)
013     C         IS = P-G-T STAB. (1,2,3,4,5, OR 6) AT T+DELTAT (NEW TIME)
014     C         SIGW = STD. DEV. IN HORIZONTAL WINDS AT T+DELTAT (M/S)
015     C         BZ = HORIZONTAL TURB. DIFF. AT TIME T (OLD TIME)  (M)
016     C       THE OUTPUT IS:
017     C         VT = VIRTUAL TIME (S)
018     C
019     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
020     C
021           SUBROUTINE ZVIRT (TIM,IS,SIGW,BZ,VT)
022     C
023     C               ***TREATMENT FOR FIRST TIME STEP (OLD TIME T=0)
024           IF(TIM.EQ.0..AND.BZ.EQ.0.) THEN
025             VT=0.
026             RETURN
027           ENDIF
028           IF(TIM.EQ.0..AND.BZ.NE.0.) THEN
029             TIM=500.
030           ENDIF
031     C
032     C            ***SPECIFY CONSTANTS FOR DRAXLER'S EQNS.
033           IF(IS.LE.4) THEN
034             C3=500.
035           ELSE
036             C3=50.
037           ENDIF
038     C
039     C               ***COMPUTE VIRTUAL TIME USING NEWTONS METHOD.
040           IF(BZ.LE.0.)BZ=1.
041           VT=TIM
042           IT=0
043         3 IT=IT+1
044           AA=SQRT(VT/C3)
045           FZ=1./(1.+.9*AA)
046           FZP=-(.45*FZ**2)/(C3*AA)
047           G=FZ*VT-BZ/SIGW
048           GP=FZP*VT+FZ
049           VT=VT-G/GP
050           IF(VT.LE.0.)VT=1.
051           CK=ABS(G/GP)
052           IF (CK.LE.10.) RETURN
053           IF (IT.GT.20)  THEN
054             WRITE (6,4)
055             STOP 'ZVIRT'
056           ENDIF
057           GO TO 3
058         4 FORMAT (2X,'VIRT. DIST. COMP. DID NOT CONVERGE IN 20 ITER.')
059           END
```

APPENDIX R
FORTRAN LISTING OF POLPRC

```
001     C  PROGRAM POLPRC.AAPRC       (VERSION  1.0)
002     C  APRIL 1985
003     C
004     C    HIGH AND SECOND HIGH MOVING AVERAGE CONCENTRATIONS ARE COMPUTED
005     C  FOR 2 POLLUTANTS, 'NSOURC' SOURCES, AND 'NRGRID' GRIDS AND/OR
006     C  'NINDR' INDIVIDUAL RECEPTORS FOR THREE DIFFERENT SPECIFIED
007     C  AVERAGING PERIODS (E.G. 3-HOURLY, 8-HOURLY, AND 12-HOURLY)
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  INPUT FILE (PRCIN_) ------------------------------------- UNIT = 9
010     C  INPUT CONCENTRATION FILES (CONC_) ----------------------- UNIT = 10
011     C                                                            THRU 14
012     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
013     C  OUTPUT (TEMPORARY) FILES -
014     C
015     C  DIRECT ACCESS FILE OF INPUT CONCENTRATIONS -------------- UNIT = 15
016     C
017     C  HIGHEST AVERAGE CONCENTRATIONS FOR EACH
018     C  GRID (MAXIMUM OF 4) ------------------------------------- UNIT = 20
019     C
020     C  SECOND HIGHEST AVERAGE CONCENTRATIONS FOR EACH
021     C  GRID (MAXIMUM OF 4) ------------------------------------- UNIT = 21
022     C
023     C  CONTRIBUTION TO HIGHEST AVERAGE
024     C  CONCENTRATIONS BY SOURCE (MAXIMUM OF 20 SOURCES) -------- UNIT = 22
025     C
026     C  CONTRIBUTION TO SECOND HIGHEST
027     C  AVERAGE CONCENTRATIONS BY SOURCE (MAXIMUM OF 20 SOURCES)  UNIT = 23
028     C
029     C  HIGHEST AVERAGE CONCENTRATION BY SOURCE ----------------- UNIT = 24
030     C  SECOND HIGHEST AVERAGE CONCENTRATION BY SOURCE ---------- UNIT = 25
031     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
032     C  PARAMETER DECLARATIONS -
033     C
034     C  KSOURC = MAXIMUM NUMBER OF SOURCES (20)
035     C  NGRID = MAXIMUM NUMBER OF RECEPTOR GRIDS (4)
036     C  NIND = MAXIMUM NUMBER OF INDIVIDUAL RECEPTORS (10)
037     C  NRECX = MAXIMUN NUMBER OF POINTS IN X-DIR FOR RECEPTOR GRIDS (25)
038     C  NRECY = MAXIMUM NUMBER OF POINTS IN Y-DIR FOR RECEPTOR GRIDS (25)
039     C  IFILES = THE PARTITIONING BY SOURCE AND POLLUTANT OF FILE ON UNIT 15
040     C         = KSOURC*(NUMBER OF POLLUTANTS)
041     C  NLINES = THE TOTAL NUMBER OF LINES IN THE CONC DIRECT ACCESS FILE
042     C         = (NRECY*NGRID + 1)*KSOURC*(# OF POLLUTANTS)*(# OF HOURS)*
043     C           *(PUFF REL RATE)
044     C         = 4040*(PUFF RELEASE RATE [IPRR])*(# OF HOURS [NHOURS])
045     C           -- ESTIMATE OF MAX. # TRACKS ON UNIVAC = 57*IPRR*NHOURS
046     C              ASSUMING OTHER PARAMETERS SPECIFIED AT MAX VALUES.
047     C           ***************************************************
048     C           *            NLINES SET EQUAL TO 300,000          *
049     C           *          (72 HOURS AT IPRR=1 FOR EXAMPLE)       *
050     C           *          (MAX # TRACKS = 4200)                  *
051     C           ***************************************************
052     C  MAYLIN = NRECY
053     C  LINES = NRECY*KSOURC
054     C  LINESI = NGRID*LINES
055     C  MAXLIN = NRECX
056     C  M2XLIN = 2*MAXLIN
057     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
058     C
059           PARAMETER
060          +  KSOURC=20,   LU=6,       NGRID=4,    NIND=10,    IFILES=40,
061          +  NRECX=25,    NRECY=25,   MAYLIN=25,  LINES=500,  LINESI=2000,
062          +  MAXLIN=25,   M2XLIN=50,  NLINES=300000
```

```
063     C
064           DIMENSION
065          +    CHI(NRECX),              CONC(NRECX,NRECY),  IAVP(3),
066          +    NPTSX(NGRID),            NPTSY(NGRID),       RSPAC(NGRID),
067          +    STAX(NGRID),             STAY(NGRID),        XIREC(NIND),
068          +    YIREC(NIND),             NSOUR(5),           IGST(NGRID),
069          +    NPTSXC(NGRID),           NPTSYC(NGRID),      RSPACC(NGRID),
070          +    STAXC(NGRID),            STAYC(NGRID),       XIRECC(NIND),
071          +    YIRECC(NIND),            XSL(KSOURC+1),      YSL(KSOURC+1),
072          +    IAVPER(3),               ITIMES(NRECX,NRECY)
073     C
074           CHARACTER*3
075          +    NAMEP(2),                NAMPC(2)
076     C
077           CHARACTER*9
078          +    ANUM1,                   ANUM2
079     C
080           CHARACTER*10
081          +    NAMES(KSOURC+1),         USENAM,       IDRUN,       IDRUNF(5),
082          +    NAMEG(NGRID),            NAMER(NIND)
083     C
084           COMMON /MOV1/
085          +  AV1(NRECX),              AV2(NRECX),              AVI1(NRECX,KSOURC),
086          +  AVI2(NRECX,KSOURC),      CONT1(NRECX,KSOURC),     CONT2(NRECX,KSOURC),
087          +  ITIMI1(NRECX,KSOURC),ITIMI2(NRECX,KSOURC),ITIM1(NRECX),
088          +  ITIM2(NRECX),           IRST(IFILES),            MLINES,
089          +  NSTEPS,                 NSOURC
090     C
091           NAMES(1)='   ALL     '
092           NSC=0
093           NFILE=0
094           NSOURC=0
095     C
096     C          ***READ IN RUN SPECIFICATION DATA FROM PRCIN
097           READ(9,*)IDRUN,USENAM,IBTZ
098           READ(9,*)IBEGC,IENDC,JFILES,NFLAG
099           READ(9,*)IXOC,IYOC,SLXC,SLYC
100           READ (9,*)NAVE,(IAVPER(M),M=1,NAVE)
101           CALL JULIAN(IBEGC,7,7,IHRB,IJULB)
102           CALL JULIAN(IENDC,7,7,IHRE,IJULE)
103     C
104     C          ***READ IN DATA FROM UP TO 5 CONCENTRATION FILES
105           DO 302 JJ=1,JFILES
106           LFU=JJ+9
107           READ (LFU) IDRUNF(JJ),IBEG,IEND,IPRR
108           IF(JJ.NE.1) THEN
109             IF(IPRR.NE.IPRRC) THEN
110               WRITE(6,105) IDRUNF(JJ)
111               STOP 'AAPRC'
112             ENDIF
113           ENDIF
114           IF(IBEG.NE.IBEGC.OR.IEND.NE.IENDC) THEN
115             WRITE (6,110) IDRUNF(JJ)
116             STOP 'AAPRC'
117           ENDIF
118           DO 1 I=1,NAVE
119         1 IAVP(I)=IAVPER(I)*IPRR
120           CALL JULIAN (IBEG,7,7,IHR,IJUL)
121           ISTART=(IJUL-1)*24+IHR
122           CALL JULIAN (IEND,7,7,IHR,IJUL)
123           ISTOP=(IJUL-1)*24+IHR
124           NHOURS=ISTOP-ISTART
```

```
125           NSTEPS=NHOURS*IPRR
126           READ (LFU) IX0,IY0,SLX,SLY
127           IF(JJ.NE.1) THEN
128             IF(IX0.NE.IX0C.OR.IY0.NE.IY0C.OR.SLX.NE.SLXC.OR.SLY.NE.SLYC)THEN
129               WRITE(6,120) IDRUNF(JJ)
130               STOP 'AAPRC'
131             ENDIF
132           ENDIF
133           X0=IX0
134           Y0=IY0
135           XEX=X0+SLX
136           YEX=Y0+SLY
137           READ (LFU) NRGRID,NINDR
138           IF(JJ.NE.1) THEN
139             IF(NRGRID.NE.NRGRDC.OR.NINDR.NE.NINDRC) THEN
140               WRITE(6,130) IDRUNF(JJ)
141               STOP 'AAPRC'
142             ENDIF
143           ENDIF
144           MLINES=0
145           IF (NRGRID.NE.0)  THEN
146             DO 2 NG=1,NRGRID
147             READ (LFU) NAMEG(NG),STAX(NG),NPTSX(NG),STAY(NG),NPTSY(NG),
148          +                    RSPAC(NG)
149           IF(JJ.NE.1) THEN
150             IF(STAX(NG).NE.STAXC(NG).OR.NPTSX(NG).NE.NPTSXC(NG).OR.STAY(NG)
151          +  .NE.STAYC(NG).OR.NPTSY(NG).NE.NPTSYC(NG).OR.RSPAC(NG).NE.
152          +  RSPACC(NG)) THEN
153               WRITE(6,140) IDRUNF(JJ)
154               STOP 'AAPRC'
155             ENDIF
156           ENDIF
157             MLINES=MLINES+NPTSY(NG)
158         2   CONTINUE
159           ENDIF
160           IF (NINDR.NE.0)  THEN
161             MLINES=MLINES+1
162             DO 3 IND=1,NINDR
163               READ (LFU) NAMER(IND),XIREC(IND),YIREC(IND)
164                 IF(JJ.NE.1) THEN
165                   IF(XIREC(IND).NE.XIRECC(NG).OR.YIREC(IND).NE.YIRECC(NG))
166          +        THEN
167                     WRITE(6,150) IDRUNF(JJ)
168                     STOP 'AAPRC'
169                   ENDIF
170                 ENDIF
171         3   CONTINUE
172           ENDIF
173           READ(LFU) NSOUR(JJ)
174           NSOURC=NSOURC+NSOUR(JJ)
175           IF(NSOURC.GT.20) THEN
176             WRITE(6,160)
177             STOP 'AAPRC'
178           ENDIF
179           DO 4 J=1,NSOUR(JJ)
180           NSC=NSC+1
181           READ (LFU) NAMES(NSC+1),XSL(NSC+1),YSL(NSC+1)
182           READ (LFU) NPLT,(NAMEP(I),I=1,NPLT)
183           IF(JJ.NE.1) THEN
184             DO 109 NP=1,NPLT
185             IF(NAMEP(NP).NE.NAMPC(NP))THEN
186               WRITE(6,151) IDRUNF(JJ)
```

```
187               STOP 'AAPRC'
188             ENDIF
189          109 CONTINUE
190             ENDIF
191            4 CONTINUE
192             IBEGC=IBEG
193             IENDC=IEND
194             IPRRC=IPRR
195             IXOC=IXO
196             IYOC=IYO
197             SLXC=SLX
198             SLYC=SLY
199             NRGRDC=NRGRID
200             NINDRC=NINDR
201             DO 106 NG=1,NRGRID
202               STAXC(NG)=STAX(NG)
203               NPTSXC(NG)=NPTSX(NG)
204               STAYC(NG)=STAY(NG)
205               NPTSYC(NG)=NPTSY(NG)
206               RSPACC(NG)=RSPAC(NG)
207          106 CONTINUE
208             DO 107 IND=1,NINDR
209               XIRECC(NG)=XIREC(IND)
210               YIRECC(NG)=YIREC(IND)
211          107 CONTINUE
212             DO 108 NP=1,NPLT
213             NAMPC(NP)=NAMEP(NP)
214          108 CONTINUE
215          C                END INPUT INFORMATION
216            302 CONTINUE
217          C
218          C***************WRITE INPUT SUMMARY TABLE*******************************
219               WRITE (LU,45)
220               WRITE (LU,46)USENAM, IDRUN,IBEG,IEND
221               WRITE (LU,90)IJULB,IHRB,IJULE,IHRE
222               WRITE (LU,91)
223               WRITE (LU,92)(IDRUNF(JJ),JJ=1,JFILES)
224               WRITE (LU,47)
225               WRITE (LU,48) SLX,SLY,X0,XEX,Y0,YEX
226               WRITE (LU,49)
227               WRITE (LU,50) NHOURS
228               WRITE (LU,51) IPRR
229               WRITE (LU,52) NSTEPS
230               WRITE (LU,53)
231               WRITE (LU,54)(IAVPER(I),I=1,NAVE)
232               DO 5 NT=1,NAVE
233               IF (IAVPER(NT).GT.NHOURS)  THEN
234                 WRITE (LU,56) IAVPER(NT),NHOURS
235               ELSE
236                 LAVE=1+(NHOURS-IAVPER(NT))*IPRR
237                 WRITE (LU,57) IAVPER(NT),LAVE
238               ENDIF
239             5 CONTINUE
240          C              ***WRITE OUT SOURCE DATA.
241               WRITE (LU,99)
242               WRITE (LU,100) NSOURC
243               WRITE (LU,101)
244               WRITE (LU,102)
245               DO 201 J=2,NSOURC+1
246                 WRITE (LU,103) NAMES(J),XSL(J),YSL(J)
247            201 CONTINUE
248          C              ***WRITE OUT RECEPTOR DATA.
```

```
249           WRITE (LU,36)
250           IF (NRGRID.NE.0)  THEN
251             WRITE (LU,37) NRGRID
252             WRITE (LU,38)
253             WRITE (LU,39)
254             WRITE (LU,93)
255             DO 203 J=1,NRGRID
256               STPX=(REAL(NPTSX(J)-1))*RSPAC(J)+STAX(J)
257               STPY=(REAL(NPTSY(J)-1))*RSPAC(J)+STAY(J)
258               WRITE (LU,94) J,NAMEG(J),STAX(J),STPX,RSPAC(J),STAY(J),STPY,
259          +    RSPAC(J)
260        203    CONTINUE
261           ENDIF
262           IF (NINDR.NE.0)  THEN
263             WRITE (LU,95) NINDR
264             WRITE (LU,96)
265             WRITE (LU,97)
266             DO 204 J=1,NINDR
267               WRITE (LU,98) J,NAMER(J),XIREC(J),YIREC(J)
268        204    CONTINUE
269           ENDIF
270     C*************************************************************************
271     C
272     C          COMPUTE STARTING RECORD LOCATION FOR EACH PARTITIONED FILE
273           IRST(1)=0
274           NFILE=NSOURC*NPLT
275           DO 10 J=2,NFILE
276           IRST(J)=IRST(J-1)+MLINES*NSTEPS
277        10 CONTINUE
278     C
279     C     COMPUTE STARTING RECORD LOCATION FOR OUTPUT FILES FOR INDIVIDUAL
280     C     SOURCES
281           IGST(1)=0
282           DO 9 NG=2,NRGRID
283           IGST(NG)=IGST(NG-1)+NPTSY(NG-1)*NSOURC
284         9 CONTINUE
285     C
286     C          DEFINE DIRECT ACCESS FILE FOR INPUT CONCENTRATIONS
287           DEFINE FILE 15 (NLINES,MAXLIN,U,IDUM1)
288     C
289     C     WRITE INPUT CONCENTRATIONS TO A DIRECT ACCESS FILE (UNIT 15)
290           IREC1=0
291           DO 15 NT=1,NSTEPS
292           NF=0
293           DO 14 JJ=1,JFILES
294           LFU=JJ+9
295           DO 14 NS=1,NSOUR(JJ)
296           READ (LFU) CLK,NSID
297           DO 14 NP=1,NPLT
298           NF=NF+1
299           IREC=IRST(NF)+IREC1
300           IF (NRGRID.NE.0)  THEN
301             DO 12 NG=1,NRGRID
302             DO 12 NY=1,NPTSY(NG)
303             IREC=IREC+1
304             READ (LFU) (CHI(NX),NX=1,NPTSX(NG))
305             WRITE (15'IREC) (CHI(NX),NX=1,NPTSX(NG))
306        12   CONTINUE
307           ENDIF
308           IF (NINDR.NE.0)  THEN
309             READ (LFU) (CHI(IND),IND=1,NINDR)
310             IREC=IREC+1
```

```
311             WRITE (15'IREC) (CHI(IND),IND=1,NINDR)
312           ENDIF
313        14 CONTINUE
314           IREC1=IREC1+MLINES
315        15 CONTINUE
316     C
317     C     DEFINE TEMPORARY DIRECT ACCESS OUTPUT FILES
318           DEFINE FILE 20(MAYLIN,M2XLIN,U,IDUM2)
319           DEFINE FILE 21(MAYLIN,M2XLIN,U,IDUM3)
320           DEFINE FILE 22(LINES,MAXLIN,U,IDUM4)
321           DEFINE FILE 23(LINES,MAXLIN,U,IDUM5)
322           DEFINE FILE 24(LINESI,M2XLIN,U,IDUM6)
323           DEFINE FILE 25(LINESI,M2XLIN,U,IDUM7)
324     C
325     C             COMPUTE HIGHEST AND SECOND HIGHEST AVERAGES FROM ALL
326     C             SOURCES (COMBINED) AND THE CONTRIBUTIONS FROM EACH SOURCE
327     C
328     C             LOOP ON NUMBER OF AVERAGING PERIODS
329     C
330           DO 29 M=1,NAVE
331           IF (IAVPER(M).GT.NHOURS)  THEN
332             WRITE (LU,70) IAVPER(M),NHOURS
333             GO TO 29
334           ENDIF
335           DO 28 NP=1,NPLT
336           KOUNT=0
337           IF (NRGRID.NE.0)  THEN
338     C
339     C             COMPUTE FOR GRIDS
340     C
341             DO 25 NG=1,NRGRID
342             NRECI1=IGST(NG)
343             DO 17 NY=1,NPTSY(NG)
344             KOUNT=KOUNT+1
345             IREC=KOUNT
346             CALL MOVAVE(NPTSX(NG),IREC,IAVP(M),IPRR,NFLAG,NGAVE,NP,NPLT)
347     C
348     C       NRECA = RECORD LOCATION IN THE DIRECT ACCESS OUTPUT FILE FOR
349     C       1ST AND 2ND HIGH AVERAGES (ALL SOURCES) AND THEIR CORRES TIMES
350     C
351     C
352     C       ITIM1(NX) AND ITIM2(NX) ARE TIMES TO THE NEAREST HOUR PAST THE
353     C       BEGINNING TIME
354             NRECA=NY
355             WRITE (20'NRECA) (AV1(NX),ITIM1(NX),NX=1,NPTSX(NG))
356             WRITE (21'NRECA) (AV2(NX),ITIM2(NX),NX=1,NPTSX(NG))
357             DO 16 NS=1,NSOURC
358     C
359     C       NRECI = RECORD LOCATION IN DIRECT ACCESS OUTPUT FILE FOR
360     C       CONTRIBUTIONS BY SOURCE AND AVERAGES BY SOURCE
361     C
362             NRECI=(NY-1)*NSOURC+NS
363             WRITE (22'NRECI) (CONT1(NX,NS),NX=1,NPTSX(NG))
364             WRITE (23'NRECI) (CONT2(NX,NS),NX=1,NPTSX(NG))
365             IF(NFLAG.NE.0)THEN
366                 NRECII=NRECI+NRECI1
367                 WRITE (24'NRECII)(AVI1(NX,NS),ITIMI1(NX,NS),NX=1,NPTSX(NG))
368                 WRITE (25'NRECII)(AVI2(NX,NS),ITIMI2(NX,NS),NX=1,NPTSX(NG))
369             ENDIF
370        16   CONTINUE
371        17   CONTINUE
372     C
```

```
C        PRINT HIGHEST AVERAGES FOR ALL SOURCES (ONE GRID)
C
         WRITE (LU,71) IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +   ,NAMES(1),NAMEP(NP),NAMEG(NG)
         DO 19 NY=1,NPTSY(NG)
         NRECA=NY
      19 READ (20'NRECA) (CONC(NX,NY),ITIMES(NX,NY),NX=1,NPTSX(NG))
         CALL GRDISP (CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),NPTSX(NG),
     +   NPTSY(NG),IHRB,IJULB,-1,LU)
C
C        PRINT TIMES OF HIGHEST AVERAGES FOR ALL SOURCES (ONE GRID)
C
         WRITE (LU,82) IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +   ,NAMES(1),NAMEP(NP),NAMEG(NG)
         CALL GRDISP (CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),NPTSX(NG),
     +   NPTSY(NG),IHRB,IJULB,1,LU)
C
C        PRINT CONTRIBUTIONS TO HIGHEST AVERAGE BY SOURCE (ONE GRID)
C
         DO 21 NS=1,NSOURC
         WRITE (LU,72)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +   ,NAMES(NS+1),NAMEP(NP),NAMEG(NG)
         DO 20 NY=1,NPTSY(NG)
         NRECI=(NY-1)*NSOURC+NS
         READ (22'NRECI) (CONC(NX,NY),NX=1,NPTSX(NG))
      20 CONTINUE
         CALL GRDISP (CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),NPTSX(NG),
     +   NPTSY(NG),IHRB,IJULB,-1,LU)
      21 CONTINUE
C
C        PRINT SECOND HIGHEST AVERAGES FOR ALL SOURCES (ONE GRID)
C
         WRITE (LU,73)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +   ,NAMES(1),NAMEP(NP),NAMEG(NG)
         DO 22 NY=1,NPTSY(NG)
         NRECA=NY
      22 READ (21'NRECA) (CONC(NX,NY),ITIMES(NX,NY),NX=1,NPTSX(NG))
         CALL GRDISP (CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),NPTSX(NG),
     +   NPTSY(NG),IHRB,IJULB,-1,LU)
C
C        PRINT TIME OF SECOND HIGHEST AVERAGES FOR ALL SOURCES (ONE GRID)
C
         WRITE (LU,83)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +   ,NAMES(1),NAMEP(NP),NAMEG(NG)
         CALL GRDISP (CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),NPTSX(NG),
     +   NPTSY(NG),IHRB,IJULB,1,LU)
C
C        PRINT CONTRIBUTIONS TO SECOND HIGHEST AVE BY SOURCE (ONE GRID)
C
         DO 24 NS=1,NSOURC
         WRITE (LU,74)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +   ,NAMES(NS+1),NAMEP(NP),NAMEG(NG)
         DO 23 NY=1,NPTSY(NG)
         NRECI=(NY-1)*NSOURC+NS
      23 READ (23'NRECI) (CONC(NX,NY),NX=1,NPTSX(NG))
         CALL GRDISP (CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),NPTSX(NG),
     +   NPTSY(NG),IHRB,IJULB,-1,LU)
      24 CONTINUE
      25 CONTINUE
         ENDIF
         IF (NINDR.NE.0)  THEN
C
```

```
C                COMPUTE HIGH AND SECOND HIGH CONCENTRATIONS FROM ALL
C                SOURCES AND THE CONTRIBUTIONS FROM EACH SOURCE AT EACH
C                INDIVIDUAL RECEPTOR LOCATION
C
                 KOUNT=KOUNT+1
                 IREC=KOUNT
                 CALL MOVAVE (NINDR,IREC,IAVP(M),IPRR,NFLAG,NINDAV,NP,NPLT)
                 WRITE (LU,75)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +           ,NAMES(1),NAMEP(NP)
                 WRITE (LU,76)
C
C        PRINT HIGH AND SECOND HIGH AVERAGES (ALL SOURCES) AND TIMES
C        AT INDIVIDUAL RECEPTORS
C
                 DO 26 IND=1,NINDR
                 CALL NUMCON(AV1(IND),ANUM1)
                 CALL NUMCON(AV2(IND),ANUM2)
                 NHRT=ITIM1(IND)+IHRB
                 IC=NHRT/24
                 IJUL1=IJULB+IC
                 IHR1=MOD(NHRT,24)
                 NHRT=ITIM2(IND)+IHRB
                 IC=NHRT/24
                 IJUL2=IJULB+IC
                 IHR2=MOD(NHRT,24)
                 WRITE (LU,77) NAMER(IND),XIREC(IND),YIREC(IND),ANUM1,IJUL1,
     +           IHR1,ANUM2,IJUL2,IHR2
              26 CONTINUE
C
C        PRINT CONTRIBUTIONS TO HIGH AND SECOND HIGH AVERAGES (BY SOURCE)
C        AND TIMES AT INDIVIDUAL RECEPTORS
C
                 DO 27 NS=1,NSOURC
                 WRITE (LU,78)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +           ,NAMES(NS+1),NAMEP(NP)
                 WRITE (LU,76)
                 DO 27 IND=1,NINDR
                 CALL NUMCON(CONT1(IND,NS),ANUM1)
                 CALL NUMCON(CONT2(IND,NS),ANUM2)
                 NHRT=ITIM1(IND)+IHRB
                 IC=NHRT/24
                 IJUL1=IJULB+IC
                 IHR1=MOD(NHRT,24)
                 NHRT=ITIM2(IND)+IHRB
                 IC=NHRT/24
                 IJUL2=IJULB+IC
                 IHR2=MOD(NHRT,24)
                 WRITE (LU,77)NAMER(IND),XIREC(IND),YIREC(IND),ANUM1,IJUL1
     +           ,IHR1,ANUM2,IJUL2,IHR2
              27 CONTINUE
                 IF(NFLAG.NE.0)THEN
C
C                    PRINT HIGHEST AVERAGES AND TIMES BY SOURCE (FOR GRIDS)
C
                    DO 30 NS=1,NSOURC
                    DO 31 NG=1,NRGRID
                    NRECI1=IGST(NG)
                    DO 32 NY=1,NPTSY(NG)
                    NRECI=(NY-1)*NSOURC+NS+NRECI1
              32    READ(24'NRECI)(CONC(NX,NY),ITIMES(NX,NY),NX=1,NPTSX(NG))
                    WRITE(LU,79)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
     +              ,NAMES(NS+1),NAMEP(NP),NAMEG(NG)
```

```
497                     CALL GRDISP(CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),
498           +         NPTSX(NG),NPTSY(NG),IHRB,IJULB,-1,LU)
499                     WRITE(LU,84)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
500           +         ,NAMES(NS+1),NAMEP(NP),NAMEG(NG)
501                     CALL GRDISP(CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),
502           +         NPTSX(NG),NPTSY(NG),IHRB,IJULB,1,LU)
503     C
504     C               PRINT SECOND HIGHEST AVE AND TIMES BY SOURCE (FOR GRIDS)
505     C
506                     DO 33 NY=1,NPTSY(NG)
507                     NRECI=(NY-1)*NSOURC+NS+NRECI1
508        33           READ(25'NRECI)(CONC(NX,NY),ITIMES(NX,NY),NX=1,NPTSX(NG))
509                     WRITE(LU,80)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
510           +         ,NAMES(NS+1),NAMEP(NP),NAMEG(NG)
511                     CALL GRDISP(CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),
512           +         NPTSX(NG),NPTSY(NG),IHRB,IJULB,-1,LU)
513                     WRITE(LU,85)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
514           +         ,NAMES(NS+1),NAMEP(NP),NAMEG(NG)
515                     CALL GRDISP(CONC,ITIMES,STAX(NG),STAY(NG),RSPAC(NG),
516           +         NPTSX(NG),NPTSY(NG),IHRB,IJULB,1,LU)
517        31           CONTINUE
518     C
519     C               PRINT HIGHEST AND SECOND HIGHEST AVERAGES BY SOURCE (FOR
520     C               INDIVIDUAL RECEPTORS)
521     C
522
523                     WRITE(LU,81)IAVPER(M),IBEGC,IJULB,IHRB,IENDC,IJULE,IHRE
524           +         ,NAMES(NS+1),NAMEP(NP)
525                     WRITE(LU,76)
526                     DO 34 IND=1,NINDR
527                CALL NUMCON(AVI1(IND,NS),ANUM1)
528                CALL NUMCON(AVI2(IND,NS),ANUM2)
529                NHRT=ITIMI1(IND,NS)+IHRB
530                IC=NHRT/24
531                IJUL1=IJULB+IC
532                IHR1=MOD(NHRT,24)
533                NHRT=ITIMI2(IND,NS)+IHRB
534                IC=NHRT/24
535                IJUL2=IJULB+IC
536                IHR2=MOD(NHRT,24)
537                    WRITE(LU,77)NAMER(IND),XIREC(IND),YIREC(IND),ANUM1,IJUL1
538           +        ,IHR1,ANUM2,IJUL2,IHR2
539           34       CONTINUE
540           30       CONTINUE
541                ENDIF
542             ENDIF
543           28 CONTINUE
544           29 CONTINUE
545              WRITE(LU,86)
546              STOP
547           45 FORMAT (1H1,4(/),T6,'***** SUMMARY OF POLPRC - CONCENTRATION FIELD
548             +S POST PROCESSOR *****')
549           46 FORMAT (1H0,T8,'USER: ',A10,/,T8,'RUN ID: ',A10,/,T8.
550             +'BEGINNING TIME (MMDDYYHH):',I9.8,2X,'ENDING TIME:',I9.8)
551           90 FORMAT(1H ,T16,'(JULIAN DAY,HOUR): (',I3,',',I2,')',15X,'(',I3
552             +,',',I2,')')
553           91 FORMAT(1H0,/,T6,
554             +'*** IDENTIFICATION OF INPUT CONCENTRATION FILES ***')
555           92 FORMAT(1H0,5(T8,A,/))
556           47 FORMAT (1H ,/,T6,'*** DOMAIN ***')
557           48 FORMAT (1H0,T8,'REGION SIZE:',F6.1,' BY ',F6.1,'  KILOMETERS',/,
558             +T8,'GRAMA COORDINATES (KM):   X=',F6.1,' TO ',F6.1,/
559             +,T34,'Y=',F6.1,' TO ',F6.1)
```

```
559        49 FORMAT (1HO,/,T6,'*** TIME PERIOD OF SIMULATION ***')
560        50 FORMAT (1HO,T8,'TOTAL HOURS OF SIMULATION:',T49,I5)
561        51 FORMAT (1H ,T8,'NUMBER OF TIME STEPS PER HOUR:',T49,I5)
562        52 FORMAT(1H ,T8,'TOTAL NUMBER OF TIME STEPS IN SIMULATION: ',T49,I5)
563        53 FORMAT (1HO,/,T6,'*** MOVING AVERAGE PERIODS ***')
564        54 FORMAT (1HO,T8,'MOVING AVERAGING PERIODS COMPUTED:',I3,'-HOURLY',/
565          +,T42,I3,'-HOURLY',/,T42,I3,'-HOURLY',/)
566        56 FORMAT (1H ,T8,'NUMBER OF AVERAGES FOR ',I2,'-HOURLY:    0  - ONLY
567          + ',I4,' HOURS OF ',/,T56,'DATA ARE PROVIDED.')
568        57 FORMAT (1H ,T8,'NUMBER OF AVERAGES FOR ',I2,'-HOURLY:',I5)
569        99 FORMAT (1H1,4(/),4X,'*** POLLUTANT SOURCES ***')
570       100 FORMAT (1HO,5X,'NUMBER OF SOURCES: ',I6)
571       101 FORMAT (17X,'NAME',5X,'X-COR',2X,'Y-COR')
572       102 FORMAT (27X,'(KM)',3X,'(KM)')
573       103 FORMAT (14X,A10,2X,F5.1,2X,F5.1)
574        36 FORMAT (1HO,/,4X,'*** RECEPTOR LOCATIONS ***')
575        37 FORMAT (1HO,5X,'NUMBER OF RECEPTOR GRIDS: ',I6)
576        38 FORMAT (30X,'X - DIRECTION',10X,'Y - DIRECTION')
577        39 FORMAT (8X,'GRID',4X,'NAME',7X,'STA X',2X,'STP X',2X,'SPACE',4X,
578          +'STA Y',2X,'STP Y',2X,'SPACE')
579        93 FORMAT (23X,2(5X,'(KM)',3X,'(KM)',3X,'(KM)'))
580        94 FORMAT (9X,I2,2X,A10,2(4X,F5.1,2X,F5.1,2X,F5.1))
581        95 FORMAT (1HO,5X,'NUMBER OF INDIVIDUAL RECEPTORS: ',I6)
582        96 FORMAT (8X,'RECEPTOR',4X,'NAME',6X,'X-CORD',3X,'Y-CORD')
583        97 FORMAT (32X,'(KM)',5X,'(KM)')
584        98 FORMAT (11X,I2,4X,A10,3X,F6.1,3X,F6.1)
585        70 FORMAT (1H1,I2,'-HOURLY MOVING AVERAGE CONCENTRATIONS CANNOT BE DE
586          +TERMINED.  ONLY ',I6,' HOURS OF DATA WERE PROVIDED.')
587        71 FORMAT (1H1,1(/),T35,'HIGHEST ',I2,'-HOURLY MOVING AVERAGE CONCENT
588          +RATIONS (MICROGRAMS/M**3)',/,T40,'FOR PERIOD: ',I8.8,
589          +' (',I3.3,',',I2.2,')   TO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
590          +'SOURCE: ',A10,3X,'  POLLUTANT: ',A3,3X,'  GRID: ',A10)
591        72 FORMAT (1H1,1(/),T25,'CONTRIBUTION BY EACH SOURCE TO THE HIGHEST '
592          +,I2,'-HOURLY CONCENTRATION (MICROGRAMS/M**3)',/,T40,' FOR PERIOD:
593          +',I8.8,' (',I3.3,',',I2.2,')   TO: ',I8.8,' (',I3.3,',',I2.2,')'/,
594          +T40,'SOURCE: ',A10,3X,'  POLLUTANT: ',A3,3X,'  GRID: ',A10)
595        73 FORMAT (1H1,1(/),T32,'SECOND HIGHEST ',I2,'-HOURLY MOVING AVERAGE
596          +CONCENTRATIONS (MICROGRAMS/M**3)',/,T40,' FOR PERIOD: ',I8.8.
597          +' (',I3.3,',',I2.2,')   TO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
598          +'SOURCE: ',A10,3X,'  POLLUTANT: ',A3,3X,'  GRID: ',A10)
599        74 FORMAT (1H1,1(/),T25,'CONTRIBUTION BY EACH SOURCE TO THE SECOND HI
600          +GHEST',I3,'-HOURLY CONCENTRATION (MICROGRAMS/M**3',/,T40,' FOR PER
601          +IOD: ',I8.8,' (',I3.3,',',I2.2,')   TO: ',I8.8,' (',I3.3,',',I2.2,
602          +')'/,T40,'SOURCE: ',A10,3X,'  POLLUTANT: ',A3,3X,'  GRID: ',A10)
603        75 FORMAT (1H1,4(/),T25,'FIRST AND SECOND HIGHEST ',I2,
604          +'-HOURLY MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)',
605          +/,T26,' AT INDIVIDUAL RECEPTORS FOR PERIOD: ',I8.8,
606          +' (',I3.3,',',I2.2,')   TO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
607          +'SOURCE: ',A10,3X,'  POLLUTANT: ',A3)
608        76 FORMAT (1HO,/,T99,'SECOND',/,T19,'RECEPTOR   X-COORDINATE  Y-COORD
609          +INATE        HIGH        HIGH       SECOND HIGH     HIGH',/,T21,'NAME
610          +      (KILOMETERS)  (KILOMETERS)  CONCENTRATION    TIME      CONCENTR
611          +ATION    TIME',//)
612        77 FORMAT (1H ,T20,A10,3X,F6.1,8X,F6.1,6X,A9,5X,'(',I3.3,',',I2.2,')'
613          +,3X,A9,5X,'(',I3.3,',',I2.2,')')
614        78 FORMAT (1H1,4(/),T21,'CONTRIBUTION BY EACH SOURCE TO THE ',I2,'-HO
615          +URLY',' MOVING AVERAGE CONCENTRATIONS (MICROGRAMS/M**3)',/,T27,
616          +'AT INDIVIDUAL RECEPTORS FOR PERIOD :',I8.8,
617          +' (',I3.3,',',I2.2,')   TO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
618          +'SOURCE: ',A10,3X,'  POLLUTANT: ',A3)
619        79 FORMAT (1H1,1(/),T24,'HIGHEST',I3,'-HOURLY MOVING AVERAGE CONCENTR
620          +ATIONS (MICROGRAMS/M**3) FROM A SINGLE SOURCE',/,T40,
```

```
     +'FOR PERIOD: ',I8.8,
     +' (',I3.3,',',I2.2,')   TOO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
     +'SOURCE: ',A10,3X,'  POLLUUTANT: ',A3,3X,'  GRID: ',A10)
  80 FORMAT (1H1,1(/),T21,'SECOND HIGHEST',I3,'-HOURLY MOVING AVERAGE',
     +' CONCENTRATIONS (MICROGRAAMS/M**3) FROM A SINGLE SOURCE',/,
     +T40,'FOR PERIOD: ',I8.8,
     +' (',I3.3,',',I2.2,')   TOO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
     +'SOURCE: ',A10,3X,'  POLLUUTANT: ',A3,3X,'  GRID: ',A10)
  81 FORMAT (1H1,4(/),T25,'FIRSST AND SECOND HIGHEST',
     +I3,'-HOURLY MOVING AVERAGEE CONCENTRATIONS (MICROGRAMS/M**3) '
     +,/,T17,'AT INDIVIDUAL RECEEPTORS FROM A SINGLE SOURCE '
     +,'FOR PERIOD: ',I8.8,
     +' (',I3.3,',',I2.2,')   TOO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
     +'SOURCE: ',A10,3X,'  POLLUUTANT: ',A3)
  82 FORMAT (1H1,1(/),T30,'TIMEE OF HIGHEST ',I2,'-HOURLY MOVING AVERAGE
     + CONCENTRATIONS (JULIAN DAAY, HOUR)',/,T40,'FOR PERIOD: ',I8.8,' ('
     +,I3.3,',',I2.2,')   TO: ',,I8.8,' (',I3.3,',',I2.2,')'/,T40,
     +'SOURCE: ',A10,3X,'  POLLUUTANT: ',A3,3X,'  GRID: ',A10)
  83 FORMAT (1H1,1(/),T30,'TIMEE OF SECOND HIGHEST ',I2,'-HOURLY MOVING
     +AVERAGE CONCENTRATIONS (JUULIAN DAY, HOUR)',/,T40,' FOR PERIOD: ',
     +I8.8,' (',I3.3,',',I2.2,'))   TO: ',I8.8,' (',I3.3,',',I2.2,')'/,
     +T40,'SOURCE: ',A10,3X,'  PPOLLUTANT: ',A3,3X,'  GRID: ',A10)
  84 FORMAT (1H1,1(/),T19,'TIMEE OF HIGHEST',I3,'-HOURLY MOVING AVERAGE
     +CONCENTRATIONS (JULIAN DAYY, HOUR) FROM A SINGLE SOURCE',/,T40,
     +'FOR PERIOD: ',I8.8,
     +' (',I3.3,',',I2.2,')   TOO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
     +'SOURCE: ',A10,3X,'  POLLUUTANT: ',A3,3X,'  GRID: ',A10)
  85 FORMAT (1H1,1(/),T16,'TIMEE OF SECOND HIGHEST',I3,'-HOURLY MOVING',
     +' AVERAGE CONCENTRATIONS (JULIAN DAY, HOUR) FROM A SINGLE SOURCE',
     +/,T40,'FOR PERIOD: ',I8.8,,
     +' (',I3.3,',',I2.2,')   TOO: ',I8.8,' (',I3.3,',',I2.2,')'/,T40,
     +'SOURCE: ',A10,3X,'  POLLUUTANT: ',A3,3X,'  GRID: ',A10)
  86 FORMAT(1H1)
 105 FORMAT(2X,' PUFF RELEASE RRATE FOR FILE ',A10,' DOES NOT MATCH',
    +          ' OTHER INPUT FIILES.')
 110 FORMAT(2X,' TIME PERIOD OFF FILE ',A10,' DOES NOT MATCH PERIOD',
    +          ' REQUESTED.')
 120 FORMAT(2X,' DOMAIN OF FILEE ',A10,' DOES NOT MATCH PERIOD',
    +          '  SPECIFIED.')
 130 FORMAT(2X,' NUMBER OF RECEEPTORS FOR FILE ',A10,' DOES NOT MATCH',
    +          ' OTHER INPUT FIILES.')
 140 FORMAT(2X,' RECEPTOR GRIDS S FOR FILE ',A10,' DO NOT MATCH THOSE',
    +          ' IN OTHER INPUT  FILES.')
 150 FORMAT(2X,' IND RECEPTORS FFOR FILE ',A10,' DO NOT MATCH THOSE',
    +          ' IN OTHER INPUT  FILES.')
 151 FORMAT(2X,' POLLUTANTS FOR  FILE ',A10,' DO NOT MATCH THOSE',
    +          ' IN OTHER INPUT  FILES.')
 160 FORMAT(2X,' GREATER THAN 200 SOURCES IN INPUT FILES.')
     END
```

```
001     C  SUBROUTINE POLPRC.GRDISP
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO WRITE OUT HIGHEST AND SECOND HIGHEST
005     C  CONCENTRATION TABLES FOR RECEPTOR GRIDS.
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE GRDISP(ARRAY,ITIMES,STAX,STAY,RSPAC,NPTSX,NPTSY,
010          +                  IHRB,IJULB,IFLAG,LU)
011     C
012           PARAMETER
013          +  NRECX=25,           NRECY=25
014     C
015           DIMENSION
016          +  ARRAY(NRECX,NRECY), DX(NRECX),DY(NRECY), ITIMES(NRECX,NRECY),
017          +  JD(NRECX,NRECY),    IHR(NRECX,NRECY)
018     C
019           CHARACTER*9
020          +  AARRAY(NRECX,NRECY)
021     C
022           ZMAX=0.
023           STPX=STAX+(NPTSX-1)*RSPAC
024           STPY=STAY+(NPTSY-1)*RSPAC
025           DO 2 NY=1,NPTSY
026           DO 2 NX=1,NPTSX
027           NHRT=ITIMES(NX,NY)+IHRB
028           IC=NHRT/24
029           JD(NX,NY)=IJULB+IC
030           IHR(NX,NY)=MOD(NHRT,24)
031     2     IF(IFLAG.LT.0)CALL NUMCON(ARRAY(NX,NY),AARRAY(NX,NY))
032           DO 3 NY=1,NPTSY
033     3     DY(NY)=STAY+NY*RSPAC-RSPAC
034           DO 4 NX=1,NPTSX
035     4     DX(NX)=STAX+NX*RSPAC-RSPAC
036           IF (NPTSX.LE.13) THEN
037           WRITE(LU,10)(DX(NX),NX=1,NPTSX)
038           WRITE(LU,9)
039           DO 5 NY=NPTSY,1,-1
040           IF(IFLAG.LT.0)WRITE(LU,11)DY(NY),(AARRAY(NX,NY),NX=1,NPTSX)
041           IF(IFLAG.GE.0)WRITE(LU,12)DY(NY),(JD(NX,NY),IHR(NX,NY),
042          +                                  NX=1,NPTSX)
043     5      CONTINUE
044           ELSE
045           WRITE(LU,10)(DX(NX),NX=1,13)
046           WRITE(LU,9)
047           DO 6 NY=NPTSY,1,-1
048           IF(IFLAG.LT.0)WRITE(LU,11)DY(NY),(AARRAY(NX,NY),NX=1,13)
049     6     IF(IFLAG.GE.0)WRITE(LU,12)DY(NY),(JD(NX,NY),IHR(NX,NY),NX=1,13)
050           WRITE(LU,10)(DX(NX),NX=14,NPTSX)
051           WRITE(LU,9)
052           DO 7 NY=NPTSY,1,-1
053           IF(IFLAG.LT.0)WRITE(LU,11)DY(NY),(AARRAY(NX,NY),NX=14,NPTSX)
054     7     IF(IFLAG.GE.0)WRITE(LU,12)DY(NY),(JD(NX,NY),IHR(NX,NY),
055          +                                  NX=14,NPTSX)
056           ENDIF
057           RETURN
058     9     FORMAT(1H )
059     10    FORMAT(1H ,/,T2,'Y(Km)'T124,' << X(Km)',T7,13(4X,F5.1))
060     11    FORMAT(1H ,T2,F5.1,13A9)
061     12    FORMAT(T2,F5.1,1X,13(2X,I3.3,'.',I2.2,1X))
062           END
```

```
001     C  SUBROUTINE: POLPRC.JULIAN
002     C  DATE: APRIL 1985
003     C
004     C    THIS SUBROUTINE IS USED TO COMPUTE JULIAN DATE AND HOUR-OF-
005     C  DAY IN THE BASE TIME ZONE GIVEN AN 8 DIGET INTEGER NUMBER
006     C  CODED WITH THE MONTH, DAY, YEAR, AND HOUR, AND THE TIME ZONE
007     C  OF THE CODED TIME.
008     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
009     C  THE INPUTS ARE:
010     C      ITIME = 8 DIGET INTEGER NUMBER CODED WITH 'MO'DA'YR'HR'
011     C      ITZO = TIME ZONE OF CODED TIME = 0(ZULU), 5(EASTERN),
012     C                          6(CENTRAL), 7(MOUNTAIN), 8(PACIFIC).
013     C      IBTZ = BASE TIME ZONE = 5(EASTERN), 6(CENTRAL),
014     C                                7(MOUNTAIN), 8(PACIFIC).
015     C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
016     C  THE OUTPUTS ARE:
017     C      IHR = HOUR-OF-DAY (01 TO 24)
018     C      JDATE = JULIAN DATE (1 TO 366)
019     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
020     C
021           SUBROUTINE JULIAN (ITIME,ITZO,IBTZ,IHR,JDATE)
022     C
023           DIMENSION
024          +    NDAY(12)
025     C
026           DATA NDAY/0,31,59,90,120,151,181,212,243,273,304,334/
027     C
028           TIM=REAL(ITIME)
029           MO=INT(TIM/10.**6.)
030           IDAY=INT(TIM/10.**4.)-MO*10**2
031           IYR=INT(TIM/10.**2.)-MO*10**4-IDAY*10**2
032           IHR=ITIME-INT(TIM/10.**2.)*10**2
033     C
034     C                ***CHECK FOR MONTH INDEX IN RANGE.
035           IF (MO.LE.0.OR.MO.GE.13)  THEN
036             WRITE (6,1)
037             STOP 'JULIAN'
038           ENDIF
039     C
040     C                ***CONVERT TO JULIAN DATE.
041           JDATE=IDAY+NDAY(MO)
042     C
043     C                ***ADJUST FOR LEAP YEAR.
044           A=FLOAT(IYR)/4-IYR/4
045           IF (A.EQ.0..AND.MO.GE.3) JDATE=JDATE+1
046     C
047     C                ***CONVERT TO BASE TIME ZONE
048           IHR=IHR+(ITZO-IBTZ)
049           IF(IHR.LE.0) THEN
050             IHR=IHR+24
051             JDATE=JDATE-1
052           ENDIF
053           IF(IHR.GT.24) THEN
054             IHR=IHR-24
055             JDATE=JDATE+1
056           ENDIF
057     C
058           RETURN
059         1 FORMAT (5X,'MONTH INDEX IS OUT OF RANGE.')
060           END
```

```
001     C  SUBROUTINE POLPRC.MOVAVE
002     C  APRIL 1985
003     C
004     C    THIS SUBROUTINE COMPUTES THE HIGHEST AND SECOND HIGHEST
005     C  MOVING AVERAGE CONCENTRATIONS, AND DETERMINES THE TIME STEP
006     C  AFTER PROCESSING START OF THE HIGHEST AND SECOND HIGHEST CONC.
007     C
008     C      AV(NX):          NEW AVERAGE FROM ALL SOURCES FOR EACH TIME STEP
009     C      AV1(NX):         HIGHEST AVERAGE FROM ALL SOURCES AT LOCATION NX
010     C      AV2(NX):         2ND HIGHEST AVE FROM ALL SOURCES AT LOCATION NX
011     C      AVER(NX,NS):     NEW AVE BY SOURCE AT LOC X FOR EACH TIME STEP
012     C      AVI1(NX,NS):     HIGHEST AVERAGE BY SOURCE AT LOCATION NX
013     C      AVI2(NX,NS):     2ND HIGHEST AVERAGE BY SOURCE AT LOCATION NX
014     C      CONT1(NX,NS):    CONTRIBUTION TO HIGHEST AVE BY SOURCE NS
015     C      CONT2(NX,NS):    CONTRIBUTION TO 2ND HIGHEST AVERAGE BY SOURCE NS
016     C      ITIM1(NX):       TIME STEP WHEN HIGHEST AVE OCCURED AT LOC NX
017     C      ITIM2(NX):       TIME STEP WHEN 2ND HIGH AVE OCCURED AT LOC NX
018     C      ITIMI1(NX,NS):   TIME OF CONTRIBUTION TO HIGHEST AVERAGE
019     C      ITIMI2(NX,NS):   TIME OF CONTRIBUTION TO 2ND HIGHEST AVERAGE
020     C
021     C      IFILES IS THE PARTITIONING BY SOURCE AND POLLUTANT OF THE
022     C      DIRECT ACCESS FILE OF INPUT CONCENTRATIONS.
023     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
024     C
025           SUBROUTINE MOVAVE (NPTSX,IRECO,IAVP,IPRR,NFLAG,NGAVE,L,NPLT)
026     C
027           PARAMETER
028          +  KSOURC=20,         NRECX=25,              IFILES=40
029     C
030           DIMENSION
031          +  AV(NRECX),         AVER(NRECX,KSOURC),    CHI(NRECX,144,KSOURC),
032          +  SUMI(NRECX,KSOURC)
033     C
034           COMMON /MOV1/
035          +  AV1(NRECX),              AV2(NRECX),              AVI1(NRECX,KSOURC),
036          +  AVI2(NRECX,KSOURC),      CONT1(NRECX,KSOURC),     CONT2(NRECX,KSOURC),
037          +  ITIMI1(NRECX,KSOURC),ITIMI2(NRECX,KSOURC),ITIM1(NRECX),
038          +  ITIM2(NRECX),            IRST(IFILES),            MLINES,
039          +  NSTEPS,                  NSOURC
040     C
041     C
042           RPRR=FLOAT(IPRR)
043           NF=L
044           AVP=FLOAT(IAVP)
045           DO 1 NX=1,NPTSX
046             AV1(NX)=0.
047             AV2(NX)=0.
048             ITIM1(NX)=0.
049             ITIM2(NX)=0.
050     1     CONTINUE
051           DO 8 NS=1,NSOURC
052           DO 3 NX=1,NPTSX
053             AVI1(NX,NS)=0.
054             AVI2(NX,NS)=0.
055             CONT1(NX,NS)=0.
056             CONT2(NX,NS)=0.
057             ITIMI1(NX,NS)=0.
058             ITIMI2(NX,NS)=0.
059             SUMI(NX,NS)=0.
060     3     CONTINUE
061           IREC=IRECO
062           IREC1=IREC+IRST(NF)
```

```
063           DO 5 NT=1,IAVP
064             READ(15'IREC1)(CHI(NX,NT,NS),NX=1,NPTSX)
065             IREC=IREC+MLINES
066             IREC1=IREC1+MLINES
067             DO 2 NX=1,NPTSX
068     2       SUMI(NX,NS)=SUMI(NX,NS)+CHI(NX,NT,NS)
069     5     CONTINUE
070           NGAVE=1
071           DO 4 NX=1,NPTSX
072             CONT1(NX,NS)=SUMI(NX,NS)/AVP
073             SUMI(NX,NS)=SUMI(NX,NS)-CHI(NX,1,NS)
074             CONT2(NX,NS)=CONT1(NX,NS)
075             ITIM1(NX)=IAVP/IPRR
076             ITIM2(NX)=ITIM1(NX)
077             IF(NFLAG.NE.0)THEN
078               AVI1(NX,NS)=CONT1(NX,NS)
079               AVI2(NX,NS)=AVI1(NX,NS)
080               ITIMI1(NX,NS)=IAVP
081               ITIMI2(NX,NS)=ITIMI1(NX,NS)
082             ENDIF
083     4     CONTINUE
084           NF=NF+NPLT
085     8     CONTINUE
086           DO 6 NX=1,NPTSX
087             DO 7 NS=1,NSOURC
088             AV1(NX)=AV1(NX)+CONT1(NX,NS)
089     7       CONTINUE
090             AV2(NX)=AV1(NX)
091             ITIM1(NX)=IAVP
092             ITIM2(NX)=ITIM1(NX)
093     6     CONTINUE
094     C
095     C       ***THE FIRST AVERAGE FOR THE FIRST AVERAGING PERIOD HAS BEEN
096     C       ***DETERMINED FOR ALL SOURCES AND ALL X-LOCATIONS
097     C       ***NEW AVERAGES ARE NOW DETERMINED FOR EACH TIME STEP
098           DO 15 NT=IAVP+1,NSTEPS
099           RNT=FLOAT(NT)
100           NF=L
101           NGAVE=NGAVE+1
102           NNT=MOD(NT,IAVP)
103           IF(NNT.EQ.0)NNT=IAVP
104           NNEW=MOD(NT+1,IAVP)
105           IF(NNEW.EQ.0) NNEW=IAVP
106           DO 12 NS=1,NSOURC
107           IREC1=IREC+IRST(NF)
108           READ(15'IREC1)(CHI(NX,NNT,NS),NX=1,NPTSX)
109           NF=NF+NPLT
110           DO 12 NX=1,NPTSX
111           SUMI(NX,NS)=SUMI(NX,NS)+CHI(NX,NNT,NS)
112     C
113     C       ***COMPUTE AVERAGE AT LOCATION NX FOR SOURCE NS
114           AVER(NX,NS)=SUMI(NX,NS)/AVP
115           SUMI(NX,NS)=SUMI(NX,NS)-CHI(NX,NNEW,NS)
116     12    CONTINUE
117           IREC=IREC+MLINES
118     C
119     C       ***COMPUTE AVERAGE AT LOCATION NX DUE TO ALL SOURCES NS
120           DO 20 NX=1,NPTSX
121           AV(NX)=0.
122           DO 21 NS=1,NSOURC
123           AV(NX)=AV(NX)+AVER(NX,NS)
124     21    CONTINUE
```

```
125     C
126     C       ***CHECK TO SEE IF AV(NX) REPLACES THE PREVIOUS HIGH AVERAGE
127             IF(AV(NX).LE.AV1(NX))GO TO 18
128             AV2(NX)=AV1(NX)
129             ITIM2(NX)=ITIM1(NX)
130             AV1(NX)=AV(NX)
131             ITIM1(NX)=NINT(RNT/RPRR)
132     C
133     C       ***FIND CONTRIBUTION BY SOURCE TO NEW HIGHEST AVERAGE
134             DO 22 NS=1,NSOURC
135             CONT2(NX,NS)=CONT1(NX,NS)
136             CONT1(NX,NS)=AVER(NX,NS)
137     22      CONTINUE
138             GO TO 23
139     C
140     C       ***CHECK TO SEE IF AV(NX) REPLACES THE PREVIOUS 2ND HIGH AVE
141     18      IF(AV(NX).LE.AV2(NX))GO TO 23
142             AV2(NX)=AV(NX)
143             ITIM2(NX)=NINT(RNT/RPRR)
144     C
145     C       ***FIND CONTRIBUTION BY SOURCE TO NEW 2ND HIGHEST AVERAGE
146             DO 17 NS=1,NSOURC
147             CONT2(NX,NS)=AVER(NX,NS)
148     17      CONTINUE
149     23      CONTINUE
150     C
151     C       ***KEEP 1ST AND 2ND HIGHEST AVERAGE BY SOURCE IF NFLAG.NE.0
152             IF(NFLAG.NE.0)THEN
153                 DO 24 NS=1,NSOURC
154                 IF(AVER(NX,NS).LE.AVI1(NX,NS))GO TO 25
155                 AVI2(NX,NS)=AVI1(NX,NS)
156                 ITIMI2(NX,NS)=ITIMI1(NX,NS)
157                 AVI1(NX,NS)=AVER(NX,NS)
158                 ITIMI1(NX,NS)=NINT(RNT/RPRR)
159                 GO TO 24
160     25          IF(AVER(NX,NS).LE.AVI2(NX,NS))GO TO 24
161                 AVI2(NX,NS)=AVER(NX,NS)
162                 ITIMI2(NX,NS)=NINT(RNT/RPRR)
163     24          CONTINUE
164             ENDIF
165     20      CONTINUE
166     15      CONTINUE
167             RETURN
168             END
```

```
001     C  SUBROUTINE POLPRC.NUMCON
002     C  APRIL 1985
003     C
004     C    SUBROUTINE NUMCON RE-FORMATS ANY FLOATING POINT NUMBER
005     C  TO THE FORM X.XXX-X OR X.XXX+X
006     C
007     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
008     C
009           SUBROUTINE NUMCON(ZNUM,ANUM)
010     C
011           CHARACTER
012          +   ANUM*9,    SIGN(2)*1,    ASIGN*1,    AA(3)*9
013     C
014           DATA SIGN/'+','-'/
015           DATA AA/'>+9.999+9','<-9.999+9','  0.000+0'/
016     C
017           IF(ZNUM.GE.1.E+10)THEN
018               ANUM=AA(1)
019               RETURN
020           ENDIF
021           IF(ZNUM.LE.-1.E+10)THEN
022               ANUM=AA(2)
023               RETURN
024           ENDIF
025           IF(ABS(ZNUM).LE.1.E-10)THEN
026               ANUM=AA(3)
027               RETURN
028           ENDIF
029           ITIME=0
030           ISIGN=1
031           IF(ZNUM.LT.0.)ISIGN=-1
032           ZNUM=ABS(ZNUM)
033           IF(ZNUM.LT.1.)GO TO 5
034           ASIGN=SIGN(1)
035           IF(ZNUM.LT.10.)GO TO 10
036     1     ITIME=ITIME+1
037           ZNUM=ZNUM/10.
038           IF(ZNUM.LT.10.)GO TO 10
039           GO TO 1
040     5     ITIME=ITIME+1
041           ASIGN=SIGN(2)
042           ZNUM=ZNUM*10.
043           IF(ZNUM.GE.1.)GO TO 10
044           GO TO 5
045     10    ZNUM=ZNUM*ISIGN
046           WRITE(ANUM,101)ZNUM,ASIGN,ITIME
047     101   FORMAT(F7.3,A1,I1)
048           RETURN
049           END
```